REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE (*MM-DD-YY*) 11/26/2008	**2. REPORT TYPE** Year III (Option 2) Final Report	**3. DATES COVERED *(From - To*)** 11/27/2007 to 11/26/2008

4. TITLE AND SUBTITLE **Multi-Sensor Information Integration and Automatic Understanding**	**5a. CONTRACT NUMBER** N00014-05-C-0294 **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Matthew Welborn Austin Eliazar	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Signal Innovations Group 1009 Slater Road Durham, NC 27713	**1. PERFORMING ORGANIZATION REPORT NUMBER** SIG.ONR.OPT2.FINAL
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Office of Naval Research ONR 251: Wade Wargo (703) 696-2574 875 N. Randolph Street Suite 1425 Arlington, VA 22203-1995	**10. SPONSORING/MONITORING AGENCY ACRONYM(S)** **11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
A

13. SUPPLEMENTARY NOTES

14. ABSTRACT
This document is submitted to ONR as a final report for the Year III effort of the ONR C2CS research carried out by the team of Signal Innovations Group (SIG), Lockheed Martin and NAVAIR (henceforth referred to as the research team) to developed technology to process general video data of interest for base and port security. This research effort has also produced a real-time implementation of the tracking and anomalous behavior detection system that runs on real-world data – either using real-time uncompressed video sensors or faster-than-real time on archived video data. Also, a key success of this research effort is that this ONR-funded C2CS effort has transitioned into multiple funded Department of Defense programs to address relevant and challenging applications in airborne persistent surveillance and airborne IED detection.

15. SUBJECT TERMS
Multi-hypothesis tracking, particle filters, anomalous behavior detection, Bayesian tracking

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr Matthew Welborn
Unclassified	Unclassified	Unclassified			**19b. TELEPHONE NUMBER** *(Include Area Code)* (919) 794-3322

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

SIGNAL INNOVATIONS GROUP, INC.

Final Technical Report

Multi Sensor Information Integration and Automatic Understanding

Contract Number: N00014-05-C0294

Submitted to

Office of Naval Research

Submitted by

Signal Innovations Group
1009 Slater Road
Suite 200
Durham, NC 27703

11/26/2008

Table of Contents

Table of Figures

Executive Summary

This document is submitted to ONR as a final report for the Year III effort of the ONR C2CS research carried out by the team of Signal Innovations Group (SIG), Lockheed Martin and NAVAIR (henceforth referred to as the research team) to developed technology to process general video data of interest for base and port security. This effort has been directed toward three primary goals:

- Develop an automated approach to real-time object tracking & anomalous behavior detection for the asymmetric threat,
- Perform this analysis while utilizing all possible data to maximize performance, but also to operate in a regime of limited training data and changing conditions, and
- Integrate informational content from multiple sources – including both sensors and human analysts.

In addition, for the final Year III of this project we have had a particular focus on transitioning the algorithms and other technology developed under this effort to airborne persistent surveillance applications that require fusion of automatic approaches with analyst to support decision making with large data sets.

A key feature of the real-time tracking and behavior detection system developed is that the algorithms *automatically learn* the statistical properties of the background environment, and are able to track new entities entering a scene. Importantly, the algorithms perform a full Bayesian analysis, in that they simultaneously maintain *multiple hypotheses* about object states (position, velocity, etc.), with this of particular relevance for tracking multiple moving entities through occlusions. The maintenance of the multiple hypotheses allows the algorithm to constantly maintain a level of confidence in inferences at any given time; this is of importance in the context of sensor management, in which multiple sensors may be sequentially deployed to minimize inference uncertainty.

The research team has also used the products from the above analysis to develop statistical models of “typical” or “normal” behavior observed in video data. In this context features are extracted from the moving entities, including spatial location, size, and vector velocity, with these evolving as a function of time. If any entity is observed subsequently to occur with low likelihood, as quantified by these statistical models, then an alert is made to a human operator. If the human deems the activity to not be anomalous, then the associated model can be refined using these new data. In this context the algorithm continually learns and evolves, as new data are observed, and the definition of “typical” or “normal” is linked to the scene under test.

The basic framework developed above has been successfully demonstrated on several video data sets, including data collected at the SIG facilities, as well as data collected at NAVAIR, China Lake. In addition, Lockheed Martin has addressed airborne data collected on a moving platform (aircraft). Lockheed Martin has also developed algorithms to address processing of video collected on multiple cameras, handoff of targets/scenes between multiple cameras, and the feasibility of performance moving object detection and tracking using a framework based on *compressive sensing* techniques. The final topic, that of a *compressive imaging* framework, is intended to address applications such as Department of Defense persistent surveillance systems where the combination of extremely high resolution sensors and large field of regard has created the situation where timely exploitation is limited by key system parameters such as sample rate, data bandwidth and sensor revisit rates.

The key results of this combined three year effort by the research team are significant and have demonstrated the effectiveness of the principled approach of using complete probabilistic modeling throughout the detection and tracking algorithm formulation and implementation. The team has demonstrated a full Bayesian object detection & tracking approach that is robust to challenging real-world lighting and background changes through the development of elegant non-parametric statistical background modeling. In addition, the use of multi-hypothesis tracking models has provided for accurate object tracking performance in the presence of dynamic and repeated object occlusions. These particular results have been critical in providing the quantitative performance gains needed to allow this C2CS technology to transition from the initial ground-based security application to the more challenging airborne persistent surveillance application space.

This research effort has also produced a real-time implementation of the tracking and anomalous behavior detection system that runs on real-world data – either using real-time uncompressed video sensors or faster-than-real time on archived video data. This implementation was made possible through the development of efficient algorithms for tracking of posterior probability distributions using particle filter techniques. This real-time implementation opens the door to more sophisticated implementations where a remote sensor can perform real-time intelligent video compression – another key enabling capability for transition to persistent surveillance applications.

The solutions developed under this effort for anomalous behavior detection are based on both instantaneous & long-term probabilistic behavior modeling. For the instantaneous case, the system *automatically learns,* in real-time, the statistical properties of the observed object behaviors and is therefore able to alert the operators to anomalous behavior activity. For the long-term behavior analysis, the algorithms allow the system to parameterize the object trajectory based on the entire observed track history and to accurately classify the object class and behavior using the parameterized models.

A final key success of this research effort is that this ONR-funded C2CS effort has transitioned into multiple funded Department of Defense programs to address relevant and challenging applications in airborne persistent surveillance and airborne IED detection. In particular, the technology developed under this program has transitioned into funded efforts to support Army persistent surveillance programs where the detection and tracking capabilities will be used to help support future IR capability for the Constant Hawk platform. Additional work has been funded to transition this technology to support development of Air Force persistent surveillance programs through funding from AFRL. Thirdly, the initial conceptual work done on the use of *relative information gain* metrics to perform sensor management under this ONR effort has been the basis of technology transition and funding through the Army Night Visions Labs (NVESD) to develop *Analyst-in-the-loop* technology to optimize the interaction of human analysts and automated classification algorithms for IED detection applications. The transition of this ONR-funded research has resulted in over $450k per year for these different programs of additional non-ONR transition funding, and additional funding is anticipated.

1 Object Tracking

A key goal of this C2CS project has been the detection of *anomalous behavior* within a video stream. This requires the modeling and analysis of the motion for each observed object within the field of view. However, underlying this analysis, there needs to be a reliable method for finding and tracking these objects, to provide trajectories for analysis. This tracker needs to be robust and reliable under any real world conditions that the camera may capture. The general solution to this challenge is still an open problem.

What we present here represents a significant improvement over the state of the art, specifically tailored for the goal of anomalous behavior detection. This section is devoted to detailing the tracking method developed at SIG to provide the relevant pose information for all of the foreground objects on the screen.

1.1 Mathematical Framework

The objective of this tracking is to detect all of the foreground objects within a scene, and provide the position and velocity information for each of these objects. Before launching into a proposed solution for this problem, it is appropriate to first define the problem more specifically. Next we will detail what this problem means from a mathematical sense. Finally, we can focus on how these definitions and equations can translate into an algorithmic solution to the tracking problem.

The first term that needs to be clarified is the definition of an object. People tend to have an intuitive understanding of this concept, but this definition can change between people, and even for a single person depending on the context. When a person is wearing a jacket, the person and the jacket may be considered a single object. Yet when the person takes off the jacket and walks away, the jacket is generally regarded as a separate object from the person. So it seems that the specification of objects is dependent on their behavior. This is especially relevant for the tracking problem, as it implies that two parts which move together can be considered a single object.

For the purposes of this problem, we specify and object as **a set of pixels in the image plane that exhibit highly correlated motion**. This definition captures much of the semantic meaning of an object, while projecting it into terms applicable to the video problem. This does imply that some objects, from our specification, may seem as two distinct parts to a human with greater scene understanding, such as two people walking together. However, for the purposes of detecting anomalous behavior, it should be clear that this distinction is irrelevant; if either object is behaving unusually, the activity will be recognized either individually or as the group.

The other concept that needs to be specified clearly is the idea of tracking itself. It seems fairly clear when we talk about an object that it has a distinct position and velocity to it. Even for a deformable object, such as a person swinging their arms as the walk, that there is a gross average motion to the object. While a situation can be envisioned where the internal motion of an object is relevant to its behavior, we have decided that such a complex model is beyond the current scope of this project. Therefore, when we talk about the pose of an object, we are referring to the gross position and velocity of the object as a whole, without regard to the configuration of its individual elements.

Given this set of definitions, we can now talk about the tracking problem in a rigorous mathematical manner. As tracking is inherently an ambiguous problem, with only indirect observations of the true state ever available, we need to talk about tracking in terms of probability. Stated as a Bayesian probability, the problem being solved is

$$p(\mu_{t+1} \mid I_{t+1}, \vec{I}_t) \qquad \textbf{(1)}$$

where μ_{t+1} is the set of all object poses (e.g. position and velocity) at time t+1, I_{t+1} is the observed image at that time, and $\vec{I}_t$ is the set of all images up to and including time t. Computing this probability directly is difficult, without knowing something about how the objects appear in the images. Therefore, we introduce a new pair of terms, A_{t+1} and $\vec{A}_t$ which are the *assignment maps* for the corresponding images. This assignment map is set of correspondences for each pixel, indicating which object is currently being observed by that pixel. If no object is observed, the pixel is assumed to be viewing the background. We can now state the problem slightly differently, conditioned on assignment maps.

$$p(\mu_{t+1} \mid I_{t+1}, \vec{I}_t) = \sum_{\vec{\mu}_t} \sum_{\vec{A}_{t+1}} p(\mu_{t+1}, \vec{\mu}_t, A_{t+1}, \vec{A}_t \mid I_{t+1}, \vec{I}_t)$$

$$= \sum_{\vec{\mu}_t} \sum_{\vec{A}_{t+1}} p(\mu_{t+1}, A_{t+1} \mid \vec{\mu}_t, \vec{A}_t, I_{t+1}, \vec{I}_t)\, p(\vec{\mu}_t, \vec{A}_t \mid I_{t+1}, \vec{I}_t)$$

$$= \sum_{\vec{\mu}_t} \sum_{\vec{A}_{t+1}} p(\mu_{t+1}, A_{t+1} \mid \vec{\mu}_t, \vec{A}_t, I_{t+1}, \vec{I}_t)\, p(\mu_t, \vec{A}_t \mid \vec{I}_t) \qquad (2)$$

where the last equation makes use of the fact that the previous estimations of pose and assignments are independent of the current image. At this point, we would like to make one approximation to this equation. Given a set of images, along with the assignment maps and object poses for those images, we assume that this data can be represented statistically as a color model, *C*, a shape model, *S*, and a trajectory model, *T*. These three models are then treated as a sufficient statistic of the previous behavior. This allows us to rewrite the equation in simplified form.

$$p(\mu_{t+1} \mid I_{t+1}, \vec{I}_t) = \sum_{\vec{\mu}_t} \sum_{\vec{A}_{t+1}} p(\mu_{t+1}, A_{t+1} \mid I_{t+1}, C_t, S_t, T_t)\, p(\vec{\mu}_t, \vec{A}_t \mid \vec{I}_t) \qquad \mathbf{(3)}$$

At this point, the equation consists of two distinct parts. The first term evaluates the current time step, while the second term forms a distribution over prior behavior. Let us focus on the first term for a moment. Through repeated application of Bayes Rule, we find

$$p(\mu_{t+1} \mid I_{t+1}, \vec{I}_t) = \alpha \sum_{\vec{\mu}_t} \sum_{\vec{A}_{t+1}} p(I_{t+1} \mid \mu_{t+1}, A_{t+1}, C_t, S_t, T_t)\, p(\mu_{t+1}, A_{t+1} \mid C_t, S_t, T_t)\;\; p(\vec{\mu}_t, \vec{A}_t \mid \vec{I}_t) \qquad \mathbf{(4)}$$

$$p(\mu_{t+1} \mid I_{t+1}, \bar{I}_t) = \alpha \sum_{\bar{\mu}_t} \sum_{\bar{A}_{t+1}} p(I_{t+1} \mid \mu_{t+1}, A_{t+1}, C_t, S_t, T_t)\, p(A_{t+1} \mid \mu_{t+1}, C_t, S_t, T_t)\, p(\mu_{t+1} \mid C_t, S_t, T_t)\ \ p(\bar{\mu}_t, \bar{A}_t \mid \bar{I}_t) \tag{5}$$

Fortunately, this long equation is unnecessarily complex. There are a number of independences that can be exercised to reduce the number of terms. First, note that the shape and trajectory models are purely predictive models of the assignments and poses of the various objects. Given the current poses, the trajectory model is irrelevant. Likewise, given the current assignment map, the shape model is not needed. This reduces the first two terms. Next, we notice that for the second and third term, there is no observation of the image. Therefore, the color model contains no information about either the assignment map or the poses. Similarly, we can remove the shape model from the third term, as without an assignment map or an image, the shape model is unnecessary. These independences allow us to rewrite the equation as

$$p(\mu_{t+1} \mid I_{t+1}, \bar{I}_t) = \alpha \sum_{\bar{\mu}_t} \sum_{\bar{A}_{t+1}} p(I_{t+1} \mid \mu_{t+1}, A_{t+1}, C_t)\, p(A_{t+1} \mid \mu_{t+1}, S_t)\, p(\mu_{t+1} \mid T_t)\ \ p(\bar{\mu}_t, \bar{A}_t \mid \bar{I}_t) \tag{6}$$

Notice that now our problem statement has clearly segmented itself into three separate and distinct modeling problems – trajectory, shape, and color – with each model building on the distribution of the previous one. There only remains one term which is ambiguous, which is the last term. However, remember that this term is referring to a sequence of time steps. If we factor out one of these time steps we find

$$\sum_{\bar{\mu}_t} \sum_{\bar{A}_{t+1}} p(\bar{\mu}_t, \bar{A}_t \mid \bar{I}_t) = \sum_{\bar{\mu}_t} \sum_{\bar{A}_{t+1}} p(\mu_t, A_t \mid \bar{\mu}_{t-1}, \bar{A}_{t-1}, I_t, \bar{I}_{t-1})\, p(\bar{\mu}_{t-1}, \bar{A}_{t-1} \mid \bar{I}_{t-1}) \tag{7}$$

which is identical to our original formulation, one step back in time. This means that we can reuse our previous distribution of the state as a prior for our new state.

This full distribution is obviously intractable to maintain completely, and has no closed form solution that can provide a parametric distribution as an answer. Instead, we take the usual approach of using a Mote Carlo sampling to represent the non-parametric distribution. However, most approaches choose to sample from the distribution at multiple points in the process. A particle filter, for instance, would perform a random sampling at each of the three models. In our proposed method, each of the three models provides a full probability distribution. We use this complete probability to provide the most accurate measure of the joint assignment map and object poses. We then sample once from the full posterior to provide a set of distinct hypotheses. Each hypothesis is then used to update separate set of prior behavior models, C, S, and T.

1.2 Trajectory Model

The first of the three core models is the trajectory model. The purpose of this model is to provide a purely predictive model, which 'blindly' evaluates the probability of a given pose, given only the previous poses of that object.

$$p(\mu \mid T) \tag{8}$$

The mathematical derivations detailed above tell us what this model is, and how to incorporate the distribution into the full posterior. However, what this math does not tell us is how to build this model. In this section, we detail how this model is constructed for the object tracking software.

Recall that the pose to be evaluated is the position and velocity of a given object. Without explicit knowledge of depth information for the camera scene, this tracking is performed in the (x,y) coordinates of the image plane. The model assumes second order motion, with the x and y motion operating independent of each other. Acceleration of the objects has been observed to be unpredictable, and is treated as a source of random noise in this model. Furthermore, we make the assumption that all noise is Gaussian distributed.

$$\begin{aligned} x_{t+1} &= x_t + x'_t + Q_x \\ x'_{t+1} &= x'_t + Q_{x'} \\ y_{t+1} &= y_t + y'_t + Q_y \\ y_{t+1} &= y'_t + Q_{y'} \end{aligned} \tag{9}$$

All of the Q terms in the above equation represent mean zero Gaussian noise. Models are updated for the next time step using the hypothesized position of the object from the current time step. Updates are performed with a traditional Kalman update.

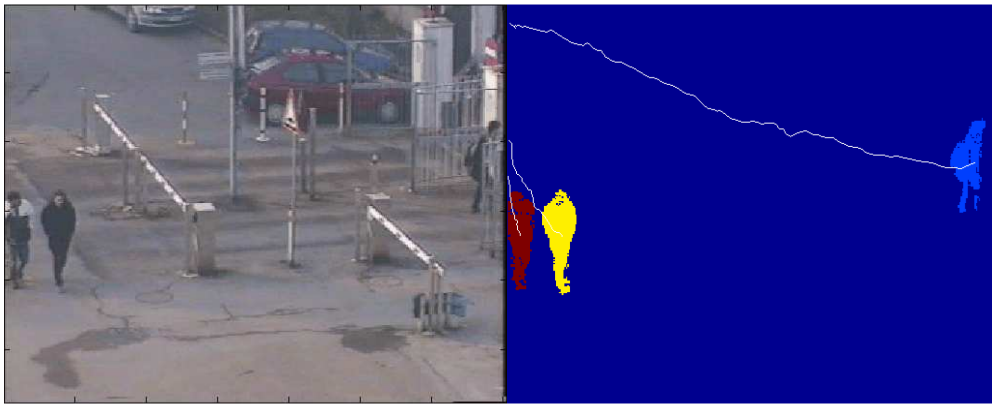

Figure 1: A visualization of the trajectory model. A point of reference for each object (e.g. center of mass) is tracked and the model maintains a non-parametric distribution of pose for each object. Object interactions are captured as this is similar to a joint particle filter over all object, but it also relaxes assumptions on data required for Kalman filters.

The resultant model closely resembles a Kalman filter in its basic form. However, it is important not to confuse the *predictive* trajectory model, which estimates the future motion of the object, with a full Kalman filter, which would provide a Gaussian posterior distribution over poses for the object. The posterior distribution over poses is maintained by the multiple sampled hypotheses, each of which results in its own distinct trajectory model. This allows the observation distribution, and thus the posterior distribution, to be non-Gaussian, and even non-parametric. Figure 1 shows an example of the trajectory model for a typical video sequence with three foreground objects.

1.3 Shape Model

The purpose of the shape model is to use a given set of object poses to predict the assignment map.

$$p(A \mid \mu, S) \tag{10}$$

This allows the tracker to incorporate valuable structural information about the objects, and thus capture the spatial dependencies between pixels. Like the trajectory model, the shape model is 'blind' in that it does not include observational evidence in its prediction.

The use of a learned shape model is highly unusual in video tracking problems like this one. Many existing methods are concerned with tracking individual key points, such as structure from motion or optical flow methods. These methods will treat a given patch of the image as locally static, and assume the entire patch moves together. As most distinctive points in an image arise from object boundaries, this is not only a false assumption, but a possibly volatile one as well.

Other methods will use explicit hand built models of specific object types, such as stick figure models of people. However, these methods are highly specific to a single domain and viewing angle. Additionally, they are incapable of handling novel object types, or alterations to existing ones, such as a person carrying a package.

These previous approaches all have their short comings, in their assumptions and their practice. Most important, however, is their inability to address the underlying problem posed by our derived mathematical formulation. None of these methods *predict* an assignment map, or evaluate the probability of a hypothesized one. Therefore, we derive a new type of shape model to accommodate our need. This model is based primarily upon the stochastic occupancy grids often used for mapping and path planning in autonomous vehicles.

First, we consider each object as its own frame of reference. As an object moves, so does the origin for that object's own coordinate system. This implies that for the most part, the components of an object tend to move together, or at least have repeatable configurations.

Second, we assume that the probability of each pixel, given the origin and the past history, is spatially independent. This allows the probability of occupancy for each pixel to be modeled independently of other pixels. This approximation is driven by the intractable nature of computing a fully covariant shape model, and is not as severe as it sounds. Better approximations may be possible, but for the purposes of this tracker, were found to be unnecessary.

Given this mobile coordinate system, we can model the probability of any given pixel belonging to this object by performing a change of coordinates from the image coordinate system into the object coordinate system. Once this remapping is established, the raw probability of occupancy can be computed using simple probabilistic frequency models. The probability of a given pixel being assigned to this object is

(# of times assigned to this object) / (# times the pixel has been observed) **(11)**

It is important to note that these two counts are maintained separately, rather than directly maintaining the probability of occupancy. This implies that when a pixel is occluded, either by another foreground object, or theoretically by a known background object, the probability of occupancy is not changed. The resultant model provides a stochastic occupancy map for each individual object, such as can be seen in the figure above.

For most deformable objects, more recent observations are more relevant than older ones. Mathematically speaking, this means that at every time step there is some chance, *p*, that something completely new occurs, and the previously observed data is no longer relevant. As this probability is cumulative, this has the net effect of exponential decay on the weight of the observations. The numerator and the denominator of the above equation are both multiplied by *(1-p)* at each time step, reducing the affect that these previous observations will have on the future. Note that we apply this decay to both terms, as compared to their quotient. This means that a point under known occlusion for several time steps will change its net probability, but the strength of this probability will drop. Therefore, the next observation will have significantly greater impact on the probability than normal.

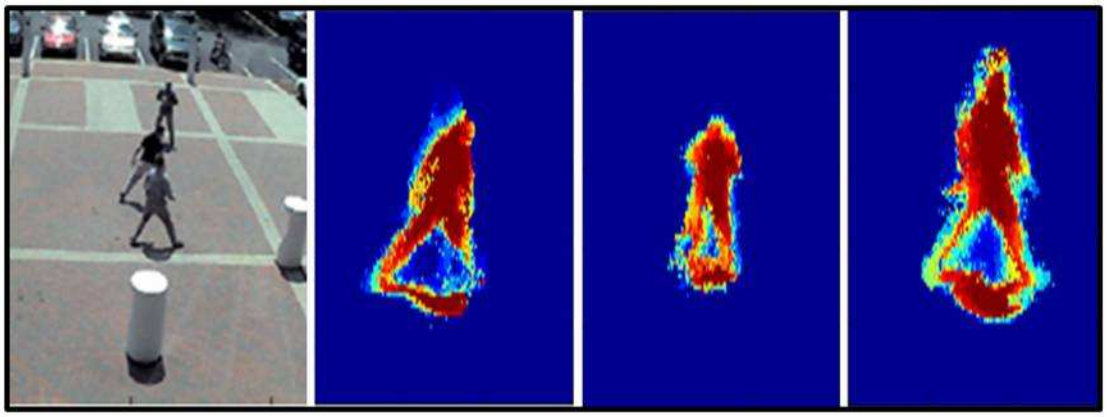

Figure 2: A visualization of the stochastic shape models. The left image shows the raw input video image. The other three images show a heat map of the probabilities output from the shape model for each of three individual objects. Note that the dynamic occlusions in the original image are not present in the stochastic shape models.

For most of our experimentation, this model is first initialized to a very low value, on the order of 10^{-5}, with the strength of the prior equal to one or two observations. This value is able to be changed in the delivered software through the graphical user interface. Note that this initialization value is not a typical prior, in that it is equivalent to previous observations, and will slowly be overwhelmed by future evidence. Instead, this value represents a background noise, a chance that something completely new might happen. Therefore, the background noise does not undergo exponential decay like the rest of the evidence, but maintains its strength regardless of the number of observations. This has the effective result of placing a minimum and maximum probability of occupancy for a given pixel.

This stochastic shape model is a powerful tool for the tracking algorithm to handle ambiguous situations. This model is crucial for allowing pixel data associations as the tracked object passes through occlusions with both static background objects and dynamic foreground objects. As pixels pass both in and out of occlusion, the shape model is able to remember the past behavior of the object's shape, and continue to expect similar assignments as the observations become less informative.

This spatial structure preserved by the shape model is also critical for resolving the identity of entire objects as they pass in front of one another. The temporal dependence on shape that is captured by this model is able to resolve the ambiguities between the tracked objects, and ensure that the tracker does not diverge, or swap identities.

1.4 Color Model

Unlike the trajectory and shape models described above, the color is an *observational* or *evidentiary* model. Given a hypothesis of the world state, the goal of the color model is to describe how well the observational evidence matches that hypothesis.

$$p(I \mid \mu, A, C) \tag{12}$$

This is a very important distinction in modeling, as the predictive trajectory and shape models have already provided us with a distribution over possible poses and assignments. There is no distribution over images to reason about, only one observation which is evaluated based on the assumptions of pose and assignment generated from a single hypothesis.

This distinction is a very important one when considering related work on previous color models. For most of these existing models, the problem is often framed as "background subtraction", where the color model is used to identify those regions of the image which are of low probability to fit the current color model. This is the same as posing the question in the inverse form, $p(\mu, A \mid I, C)$. As can be seen from the equations at the start of this chapter, this formulation obviously mishandles the evidence, and disregards the prior probabilities of alternative hypotheses. At the least, this approach regards each hypothesis as equally probable in the prior.

While the formulation used by alternative models is not solution we need for this approach, the underlying color models could still provide a good model of the behavior of the background colors. The simplest of these models are usually some derivation of interframe differences. In this model, a single canonical image is constructed as the 'true' image that is expected to be observed, if no foreground objects were present. This presents a single intensity value for each pixel, and any deviation from this value is treated as noise. The noise models across the image are identical, and generally treated as Gaussian.

These interframe models tend to be fairly brittle, especially in dynamic environments. Many elements of the world, such as trees and reflections, tend move slightly in the background, without providing a meaningful object to track. Even in indoor static situations, slight vibrations and lighting changes can produce severe amounts of false positives. These increase over time, and generally make interframe models unusable for any decent amount of time without regular resetting, generally involving a human operator.

To address many of these small changes in a background environment, many researchers treat the background color models as multimodal. Here, the each pixel in the world is assumed to have a finite number of 'states' that it can achieve. Each of these states corresponds to a separate mean for a probability distribution, and occasionally maintains a separate variance as well. As with interframe models, the noise models for these distinct modes are represented as Gaussians as well, resultant in a form called Gaussian Mixture Models (GMMs). Often, each of these Gaussian models is weighted with a relative probability, to allow some states to occur more often than others.

While these methods are better at handling foliage and similar dynamics, they are still prone to systematic failure with sudden or unusual changes. This is largely due to their inability to react quickly to new behaviors in the distribution, and their inherent complexities in the data association. For predicting a new color, a mixture of Gaussians works well, as a summation over all of the possible states (*i.e.* Gaussian modes) that the mixture contains. However, when updating these models, the precise state to be updated needs to be known. Updating GMMs is notorious for the under-specification of this data association problem. For instance, one mode can be hidden within another, when the modes overlap in their first standard deviation, and the entire non-symmetric distribution will be over generalized with a single wide variance Gaussian. Also, the addition of new modes is an undefined problem in an online algorithm. These complexities and local optima tend to greatly reduce the effectiveness of GMM color models.

The color model used in this project consists of a new multi-hypothesis model, with similarities to both of these previous methods. We first describe this method in relation to a background color model. Afterwards, we show how this model is adapted to a color model for the distinct foreground objects being tracked.

In this color model, each pixel maintains a history of the N most recent observations of the background at this point. As with the shape model, if a pixel is occluded by a foreground object, no observation is recorded, and the model is not updated for that pixel in any way. Furthermore, we treat the color model for each pixel as independent of the surrounding pixels. This significantly reduces complexity of the model, while not resulting in a significant loss of information.

Given this history of N previously observed values, we can treat each of these previous time steps as a distinct hypothesis of the 'true' color emanating from the real world. The current pixel intensity is allowed to deviate from this hypothesized according to our noise model, but each hypothesis generates its own probability value. These probabilities are then combined with a weighted sum to generate a total probability for the observed intensity.

$$p(i_t \mid i \in background, C) = \sum_{n=1}^{N} p(i_t \mid i_{t-n}, i_t \equiv i_{t-n})\, p(i_t \equiv i_{t-n}) \qquad \textbf{(13)}$$

where i_t is the intensity of a specific pixel at time t, and $i_t \equiv i_{t-n}$ implies that i_t and i_{t-n} are both drawn from the same underlying state in the real world. For our purposes, we treat $p(i_t \equiv i_{t-n})$ as a uniform constant value, though it would be reasonable to treat this as a time dependent function.

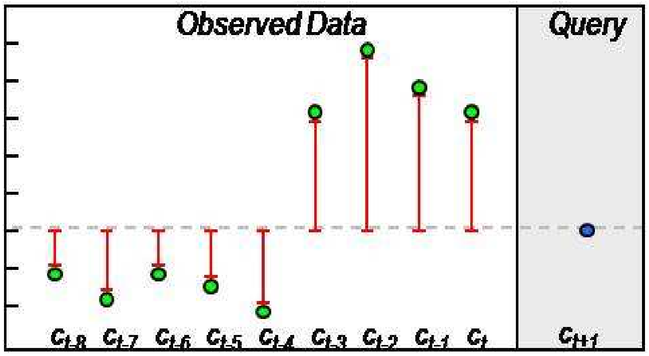

Figure 3: Typical values of the color model for a particular pixel in a scene showing the *N* previous observed values and the current pixel value as a "query" relative to the multi-hypothesis color distribution.

This relatively simple model has several desirable properties. First, it treats the distribution as an average over probabilities from several frames, rather than more generic models such as a GMM which treat the distribution as a single probability from the average intensity over several frames. This allows for a much more nimble distribution, capable of quickly adapting to new changes in the environment. As soon as a given value is recognized as being a part of the background (through other elements of the tracker, or simply as a sampled hypothesis), the model immediately recognizes other similar values as being of higher likelihood. Models which explicitly estimate the mean of a distribution will continually lag behind the true current mean. This is especially problematic for gradually changing appearances, such as shadows.

Additionally, this model has a less parametric structure to its probability distribution, due to the large number of weak modes in the model. In effect, this model asymptotically approaches a non-parametric histogram model as the length of the temporal history is increased. If a single value is observed repeatedly in the data, the model can be seen as maintaining multiple modes at that point, all with the same mean. This has the net effect of treating that mean as having an increased weight, compared to other points in the distribution.

Finally, the formulation of the model makes the data association problem irrelevant to the updating of the model. While the prediction still integrates over all possible data associations of the current observation to previous ones, this association is not used during the update stage. A new hypothesis is created, regardless of the value of this new observation, removing the pitfalls of missed associations or values falling in the tails of the existing distributions.

It has become canonical in computer vision to treat color noise models as Gaussian distributions. However, in our own experimentation, we have found these models to prove a poor fit to the observed behavior of our cameras. Observing a completely stationary object with consistent lighting, an empirical histogram of the observed intensities for a given pixel were distinctly more peaked than a Gaussian, and in fact were more accurately fit by a double exponential distribution, $\lambda e^{-2\lambda \cdot |x-\mu|}$. It is this distribution that we use for the noise distribution in this color model.

While this method has been described until now in terms of the background color model, the same basic model is applicable to the foreground objects as well. As foreground objects are moving between frames, it becomes necessary to move the corresponding color model as well. This can be achieved through a change of origin, in the same fashion as the stochastic shape models described earlier. However, due to the dynamic nature of a foreground object, including internal motion, it is necessary to allow greater leeway in the distributions for foreground color models. This is achieved by a weighted combination of the probability output from the color model with a strong prior model for the color distribution. Essentially, we maintain a reasonable chance that the colors previously observed at a given location will not be correlated with the currently observed color. Therefore, for some non-trivial amount of the time, we can expect the probability of the current observation to be uniformly distributed, and thus equal to 1/256, the total range of intensities returned by our camera. Intuitively, this background probability can be seen as placing a floor on the possible probability of any single pixel observation, and preventing a single unlikely event to ruin the total evaluation of the hypothesis.

One final extension to the color model is implemented, to handle changing lighting conditions. In outdoor scenes, the environmental lighting can change rapidly and drastically even over short periods of time. Additionally, automatic gain control systems are often slow to respond, in accurate, or respond to the inclusion of a new foreground image rather than an actual lighting change. To handle these lighting changes, each frame estimates the total change in lighting observed at the current frame. This is done using interframe changes in the intensity of the background pixels only. Since the background pixels are assumed to be spatially independent with symmetric noise, the average interframe changes can be assumed to provide a fairly low variance on the true lighting change between the frames.

The intensity values reported by the camera are known to be an exponential function of the number of photons detected. As the illumination of a scene increases, all pixels do not vary their intensity linearly, but rather in the form of $i_t = i_{t-1}^{1+\alpha}$ for some lighting change of α. Therefore, the known background pixels are used to estimate the best fit for α in a given time step. This lighting change is then used, cumulative with the previously observed lighting changes, as an estimate of the lighting conditions in the next time step to correct that image. As these lighting corrections are done without any dependence on observations or hypotheses for the current time step, they can be applied during the preprocessing step, to help adjust the virtual gain of image, and provide a much lower amount of noise in the color models, by mitigating the systematic illumination noise. This method provides drastically improved results for many situations, particularly at dawn and dusk, and during partly cloudy conditions.

1.5 Object Segmentation & Data Association

We have described the three models – color, shape and trajectory – which represent the core of the tracking engine. These three models are combined to give the probability of a set of poses and an assignment map, based on the evidence available. However, these models merely provide a probability distribution. Maintaining this probability distribution over poses and assignments, and updating the models based on this joint distribution, is not a straight-forward process. Each distinct point within the joint distribution will update the three models in a different way.

In the method developed for this project, this posterior distribution is represented by a finite set of distinct hypotheses, each one representing a different state of the system. This state space is very large, especially as the number of objects increases. However, it is important to note that the interaction between objects is fairly weak. The estimation of the pose for one object is rarely affected by the estimation of a second object. Therefore, we reduce the dimensionality of the state space by allowing each object to maintain its own independent set of hypotheses. Each object then maintains a distinct set of hypotheses, each one consisting of a unique trajectory, shape, and color model, as updated by a single sample of the joint pose and assignment map for this object.

When these objects do interact, it is often sufficient to use the most likely estimation for one object to evaluate the distribution for the other, and vice versa. This means that in certain situations, not all of the evidence is used, such as when two objects pass by each other. It is possible that in this situation, each object now has a multimodal distribution, representing the uncertainty in the resultant data association, possibly leading to the identities of the two objects to be swapped in some hypotheses. Intuitively, it would seem that the joint distribution between these two objects would help resolve the ambiguity. However, empirical evaluation of the resultant probability distributions in such a situation has found that the probability distributions demonstrate only a limited change when using a joint distribution of the two objects over an independent one. The increase in computational efficiency and the significant resultant increase in available hypotheses for each object, more than justifies this approximation of independence between the objects. A hybrid model, which models objects as a joint distribution solely when they are directly dependent, has been developed, but is not yet mature enough for use in this application.

So far, we have concentrated on the process on tracking objects. However, initial detection and segmentation of the objects is crucial, in that it provides something to be tracked. This object segmentation relies largely on the background color model, as the shape and pose information for an unobserved object remain constant priors. The shape prior is a very low uniform probability across the entire image, on the order of 10^{-5}, representing our complete lack of distinguishing information about the object. The foreground color model is likewise uninformed, and is also a uniform distribution across the color space for each pixel. These two models are then combined to produce the likelihood of a new object appearing at any pixel.

$$p(a_{x,y} = newobject \mid C, S, i_{x,y}) = \alpha \cdot p(i_{x,y} \mid C, a_{x,y} = newobject)\, p(a_{x,y} = newobject \mid S) = \alpha \cdot \tfrac{1}{256} \cdot 10^{-5} \quad \textbf{(14)}$$

Here, $a_{x,y}$ represents the assignment of a specific pixel, and $i_{x,y}$ represents the observed intensity value at that pixel. The α term continues to represent a normalization term, to ensure that the probabilities of all possible assignments add to 1. This implies that the likelihood of each pixel belonging to a completely new object is a constant. To determine the full probability of this occurrence at a given pixel, we need to determine value of the parameter α. This is equivalent to normalizing all of the likelihoods of assignment at that pixel – all of the foreground objects, the background, and this possible new object. This hypothesis of a new object will seem plausible only if the other assignments are themselves highly unlikely. This is a probabilistic form of the old axiom, that when you eliminate the impossible, whatever remains, no matter how improbable, must be the truth. If no other object has a strong claim on this pixel, and the background color model seems unlikely, we have no alternative but to entertain the possibility that a new object has appeared.

Given a set of pixels which are part of new objects, we still have to decide how to segment them. Given the vast array of possible segmentations, we rely on spatial dependencies to help differentiate these various objects. Using a simple rule of spatial connectivity, we gather the pixels into distinct objects, under the assumption that two simultaneously appearing objects appearing with spatial overlap is independent of the probability of one object occurring there, and thus is on the order of our shape prior, 10^{-5}.

These methods of tracking and segmentation are, of course, capable of error. This is largely due to the limited number of hypotheses that can be maintained at each time step. This can lead to erroneous assignment maps, which then have the potential to cause the tracker to diverge. Even beyond the quality of the hypotheses maintained, there are natural occurrences in the world which can lead to changing assignment maps, such as a person getting out of a car, resulting in a single object becoming two. To handle these corrections, and maintain correct tracking behavior, we recognize in our models the possibility of four events which cause significant changes in the assignments of certain pixels. These are background objects, negative objects, object splits, and object merges. Each of these is described in detail below.

Background objects are exactly what their name suggests. These are groups of pixels which are objects in their own right, but which exhibit no perceptible motion. For all purposes of tracking, these objects are a part of the background, and the tracker would best be served by not spending resources tracking and maintaining these stationary objects.

Background objects occur when an object has entered the scene, and exhibited coherent motion, sufficient to meet our definition of an object. However, this object eventually comes to rest, such as a car pulling into a parking space, and then does not move for an extended period of time. This exact period of time can be defined by the user, and is typically set to something on the order of 20 seconds to a minute, though it can be as long as is appropriate to the scene. These objects are easily recognized as being completely stationary over this time, and the tracker makes the deliberate decision to place this object into the background. The object is removed from the list of tracked objects, and its appearance is added into the background color model as appropriate. Should the object start moving again after being designated as a background object, it will be detected as a new object, and will be tracked as such.

The corollary to background objects is negative objects. These are not true objects by our definition of the term, but rather groups of pixels which would appear to belong to neither the background nor to any existing object. These are typically observed when an object that was previously considered a section of the background begins to move, such as a car leaving a parking space. The object itself is detected as a new object, and is properly tracked. However, the area in the image that the object previously occupied has changed. Instead of observing the object at these pixels, as the color model is used to, we instead observe what was behind the object, which is expected to look significantly different. This means that by our segmentation priors, we have a potential object at this location. However, remember that our definition of an object requires correlated motion. If the 'object' remains still, it cannot be considered part of the foreground. Therefore, any new potential object is held in limbo for several frames, until its proper segmentation is determined, and the 'object' can be determined to be part of the foreground or the background. Once motion is apparent, and the object is recognized to be part of the foreground, the tracker adds this to its list of objects, and begins reporting the trajectory of this new object. This greatly reduces the number of false positives maintained by the system, and allows the tracker to concentrate resources on those objects which are truly of interest.

As we have alluded to earlier, not all parts of an object will continue to move together. A person can set down a bag or take off a jacket, or a car can let out a passenger. Objects in the real world separate all the time, and it is important for the tracker to be able to recognize these events, and track the distinct portions independently. Therefore, we have built in to the tracker the possibility of an object to split into two or more pieces. Without this capability, there is an implicit assumption the implementation that assignments cannot change, an assumption which is obviously false.

An object split is based primarily on consistent differentiated motion of two sections of a single object. However, as internal motion and self occlusion can greatly confuse the process, in our implementation we also use spatial separation to aid in segmentation of the object into its two parts. The basis of the split model is to continually hypothesize that the object has separated into two distinct subcomponents, each with its own distinct motion. Under this hypothesis, the object is allowed to have two different velocities, either of which could be used to describe the motion of a given pixel in the object. These motion pairs are examined for support using the standard likelihood model, $p(I_{t+1} \mid \mu_{t+1}, A_{t+1}, C_t)\, p(A_{t+1} \mid \mu_{t+1}, S_t)\, p(\mu_{t+1} \mid T_t)$. If the secondary motion can show consistent support over several time steps for a coherent trajectory, as captured within T_t, while still matching the observed images, the dual object hypothesis will develop a high likelihood. When combined with the relatively low prior probability for the event of an object splitting in two, this probability of separation can be compared to the probability of the object remaining whole. New hypotheses can be sampled from this relative probability to create an actual object split. When an object is split, pixels are assigned to subparts according to spatial connectivity, and each subpart is then assigned to one of the two objects based on its predominant motion. This allows some internal deformation, without unnecessarily complicating the models, yet enforcing that the motion of all of the pixels in an object continue to exhibit highly correlated motion.

An object merge is merely the inverse problem: two objects now behaving in a highly correlated fashion, and in effect becoming a single object. Consequently, the model for detecting such a merge is handled almost identically, but with the goal of detecting a single coherent motion to consistently describe the motion of both objects. In both the case of split and merge, the assignment of object identities, which object maintains the consistent identity, is largely arbitrary, as the models are initially identical regardless of the choice of identity assignment.

1.6 Empirical Evaluation

Over the course of this project, the algorithms have undergone continuous testing to evaluate the performance and justify the use of our models. The most relevant of these tests are presented here, to demonstrate the capabilities and strengths of the methods used in this project. Unless otherwise stated, all of the experiments described were run on data collected at 15 frames per second, providing uncompressed 640x480 Bayer pattern images with 8 bit intensities. Experiments were performed on prerecorded images to ensure data invariance with reproducible results, though all processing was done in an online manner indistinguishable from a live video feed.

The first experiments we present were performed to establish the performance of the color model, both objectively as well as in relation to Gaussian Mixture Model approaches. In these experiments, video was collected of a large tree under windy conditions. The goal of the experiment was to determine the accuracy of the color model under adverse background conditions, with changing color values where the leaves of the tree move.

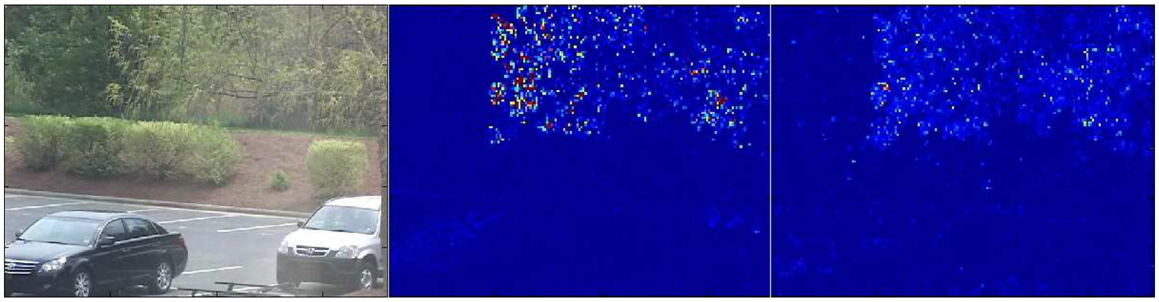

Figure 4: Left: raw video input. Center: the probability of the color given a GMM color model for the background. Right: the probability of the color given the multi-hypothesis color model for the background.

The figure above shows a raw input image from the sequence on the left, with a heat map of the individual probabilities of that color, given the background color model, on the right, with blue being the most probable and red being the least probable. Also shown for comparison is the probability map produced by an optimized version of the popular Gaussian Mixture Model method proposed by Stauffer, Grimson et al. The multi-hypothesis color model can be seen to handle the motion of the leaves very smoothly, and easily predict the observed values as high probability. The GMM method has a much more difficult time dealing with this method, despite being specifically designed for this type of motion, and tuned to this specific scenario. Using these two color models in the tracking algorithm, the GMM model produced as many as seven false objects at a time within the leaves of the tree. Furthermore, the GMM model initially failed to detect a person walking behind the tree, due to overly large variances on the models, and later only detected the head and the legs as two separate objects. The multi-hypothesis color model, by comparison, detected the person walking behind the tree as soon as a single leg is visible, and returned absolutely no false objects.

A number of similar experiments were performed to evaluate the multi-hypothesis color model, encompassing 16 minutes worth of footage from 8 different scenes involving potentially difficult backgrounds. These scenes included foliage, reflections on parked vehicles, and rain. During these 16 minutes of test footage, there was only one false object detected, as a particularly strong gust of wind moved a tree violently. This object disappeared almost immediately, as the tree settled back to its previous dynamic state.

A separate set of experiments were designed to test the capabilities of the color during drastically changing lighting conditions. Data was collected late in the day, with rapidly changing cloud cover. In this test set, the lighting conditions can be seen to change from bright illumination, resulting in distinct shadows and super-saturation of sections of the image, all the way to fully overcast, with no apparent shadows and significant under-saturation, and back again all within the space of 40 seconds. The multi-hypothesis color model performed equally well on this test set, exhibiting no false objects over the entire course of the 20 minutes of test data.

We have seen the performance of the color model under empirical evaluation. Similar testing was performed for both the shape and trajectory models, but these results are less intuitive on their own. In particular, there are no alternatives in existing literature to the stochastic shape model to provide for comparison. Therefore, we choose to present the results of the tracking system as a complete whole, under various conditions, to demonstrate the capability of these two models, along with the rest of the tracking system.

These experiments were carefully tested on a large number of 'natural' scenes, or scenes that commonly occur without any interference on our part. The data collected outside of the entrance to the SIG offices, with a camera pointed down at the parking lot from a second story window, in the same type of configuration used by many existing security cameras. This area has been observed to have similar types and density of traffic as a canonical scenario indicated by the project manager. Many vehicles can be seen throughout the videos, coming and going at various distances from the camera, along with significant pedestrian traffic originating both from the vehicles and from off screen. Over a ten minute period, it was typical to observe an average of 28 objects enter the scene, with as many as seven separate objects appearing at a time. Most of these objects traversed the entire field of view, with vehicles taking a few seconds, and most pedestrians taking an average of 10-20 seconds. The size of objects varied significantly, from as many as 15,000+ pixels on target, down to only six pixels observable for an object.

One hour of test data was used in a blind test of the algorithm, in distinct ten minute clips, with a human providing ground truth for the data. Over the course of this test, the algorithm demonstrated an object detection rate of 99.8%, with approximately 2% false positives. The tracks of the objects were examined, and the tracking was determined to diverge an average of .05 times for any given minute that an object was on screen. Data associations were evaluated by the number of dynamic occlusions that were observed, where one foreground object significantly overlapped in the image plane with another foreground object. In these situations, the resulting objects were determined to maintain a proper consistent identification 98.5% of the time. It is interesting to point that of all of these observed dynamic occlusions, 1.3% of the original events were not included in the evaluation, due to the human analyst being uncertain of the true associations. Over the course of these videos, using a 2 GHz Pentium M processor, the frame rate of processed video never dropped below the 15 frames per second available from the camera.

The performance of a complete video tracking system consists not only of the algorithms, but also on many aspects of the system hardware. The cameras, processing, and other equipment can impact the performance of the tracking system and it can therefore be useful to test the effects of varying some of the system variables. Of particular note were the effects of the image resolution and video frame rate on the quality of the tracks.

For one set of experiments, we down-sampled the prerecorded video via averaging, to simulate the effects of a lower resolution sensor with the same distance and field of view. We used down-sampling factors of two, four, eight, and sixteen, producing video images as small as 40x30 pixels. Predictably, the tracking system continued to have difficulty tracking objects with less than six pixels on target, and with apparent motion of less than 0.25 pixels/frame. With these exceptions, the resultant poses appeared very similar, with uncertainty and variance increasing roughly linearly with the down-sampling factor used.

For our next set of experiments, we artificially varied the frame rate of the camera, feeding the data to the algorithm at seven and a half, five, two, and one frames per second. Once again, the tracker performed remarkably similarly, with uncertainty and variance in the tracks seeming to grow slightly faster than linear with the reduction in frame rate. Data associations degraded slightly, with persistent identification dropping as low as 95% for 1 frames-per-second (fps). Track divergence maintained relatively steady through .06 per minute at 2 fps, and then dropped off to 0.11 per minute at 1 fps. The other significant difference appeared in particularly fast moving objects, which are on screen for less than 5 frames, which appear very spatially distant between frames. These objects would sometimes not be detected as foreground objects, or appear as several different objects.

Parameters of the observed scene were also varied, to test the ability of the tracking system to handle greater object density, approaching the level of crowds. In these staged scenarios, a large number of people were asked to walk through the camera's field of view, to augment the naturally occurring activity in the scene. Two scenarios of ten minutes each were collected for blind testing.

During these experiments, as many as eleven objects can be observed on screen at a time. Consequently, the number of dynamic occlusions increased significantly, often with four or more objects in significant overlap at a time. During this time, the rate of divergence increased to .097 per minute for each object, and the percent of correct data associations dropped to 91%. The percent of objects detected was 100%, with a false positive rate well below 1%. During this entire period, the lowest that the frame rate ever dropped to was 12 fps, and then quickly recovered above 15 fps within a few frames.

2 Behavior Modeling

Of significant importance for current and anticipated DoD activities, the ISR system developed under this C2CS effort is designed to detect asymmetric threats, with the goal of recognizing unusual behavior or activities. The technologies and systems developed under this effort are designed for semi-automated scene awareness, with the objective of recognizing behavior that appears atypical (e.g. atypical object motion, and dynamic characteristics of people and vehicles). To that end, SIG has leveraged our previously developed technology to develop second-generation methods to adaptively learn the statistics of dynamic object behavior in video, while focusing on defining system requirements for sensor deployment by using field data (vs. highly controlled indoor data).

2.1 Problem Statement

In contrast to rule-based event detection algorithms, our anomalous behavior detection algorithm is designed to learn normal patterns of behavior through observation and report anomalous behaviors as they occur. Our tracking algorithm produces poses, instantaneous positions and velocities, for each object every frame. A pose is an instantaneous second order description of an object's behavior. Given many pose observations, we can model a probability density over the space of poses. A pose of high probability given our model is normal, while a pose of low probability given our model is anomalous.

2.2 Behavior Models

We present in this section results for our anomalous behavior detection work using both instantaneous object state estimates (velocity and position) as well as results for behavior class detection using full object trajectories.

2.2.1 Instantaneous observations

Given a set of pose observations, we create a histogram density estimate over the space of poses smoothed with a truncated Gaussian. Mathematically, a pose is a position in a 4-dimensional space. Let x and y be the horizontal and vertical positions respectively. Let x' and y' be the horizontal and vertical velocities. We define the object pose μ as

$$\mu = \begin{bmatrix} x \\ y \\ x' \\ y' \end{bmatrix}. \tag{15}$$

We treat each pose as an independent observation, and we assume that the dimensions of the pose are independent.

Let w and h be the image width and image height. Let x'_{min} and x'_{max} and y'_{min} and y'_{max} be arbitrary bounds on the horizontal and vertical velocities. We build a $w \cdot h \cdot (x'_{\max} - x'_{\min}) \cdot (y'_{\max} - y'_{\min})$ bin histogram with one bin for each integer in the sub-space $[0,w]\times[0,h]\times[x'_{\min}, x'_{\max}]\times[y'_{\min}, y'_{\max}]$. Poses are rounded to the nearest bin.

Initially, an equal amount of weight s_0 is placed in each bin of the histogram. This has the desired effect of treating all poses as equally likely when no observations have been added to the behavior model. In effect, we are imposing a uniform prior on our histogram density estimate. Increasing or decreasing the amount of weight has the effect of increasing or the decreasing the strength of this prior.

Let σ^2 be the variance and $[-r:r]$ be the bounds of the truncated Gaussian we use for smoothing the histogram. When we observe a new pose μ, we compute the weight $s = \exp\left(-\|p-\mu\|^2 / \sigma^2\right)$ for each point p in the space $[x-r, x+r]\times[y-r, y+r]\times[x'-r, x'+r]\times[y'-r, y'+r]$ and add it to p's bin in our histogram. This has the effect of smoothing the histogram with a Gaussian. The resulting probability density function is similar to that obtained via Gaussian kernel density estimation, but without the overhead of maintaining a list of observations. It is not necessary to normalize each weight individually, since σ^2 is always the same. To find the probability of pose μ, we find its bin and divide the value in that bin by the sum of the values in all of the bins.

2.2.2 Full Object trajectories

In addition to the real-time anomalous behavior detection work described above, we have also developed models of the behavior of humans and vehicles in video sequences using their motion trajectories. In this work, the trajectories of objects are automatically extracted are sampled based on arc-length. The link between these trajectories and semantic labels of behavior are established using a boosting-based learning strategy. The effectiveness of the approach is demonstrated using behavior recognition and behavior prediction. In behavior prediction, the posterior probability that a test object will attain a known behavior at a future time is computed, conditioned on data available until the present time.

For traditional classification approaches, it is preferable to compute features that have the same number of components across samples for designing classifiers. Hence raw trajectories extracting using our tracking results detailed above are not suitable as features for training classifiers because the length of trajectories varies across samples. In order to obtain invariance to time-parameterization, the extracted trajectory is re-sampled according to its arc length. The trajectory $r(t) = [x(t), y(t)]$ is thus represented as a function of arc-length as $r(s)$. The trajectory is thus parameterized according to its arc length, $r(s) = [x(s), y(s)]$. It is easy to show that the arc length parameterization thus obtained makes the re-sampled trajectory invariant to time sampling.

We assume a supervised setting to categorize objects into semantically relevant classes. The training set consists of the trajectory features $x_1, \dots, x_{N_s}$, where x_n, $n = 1, \dots, N_s$ and N_s is the number of training samples; and labels $y_1, \dots, y_{N_s}$, where $y_n = (y_{n1}, \dots, y_{nJ})$ is a J-dimensional binary-valued vector. The value J is the number of classes of activities of interest; $y_{nj} = 1$ indicates that the nth training sample belongs to class j.

Boosting has proved to be an effective classification strategy that iteratively improves weak classifiers. Starting with a weak learner, a boosting algorithm generates weak hypotheses that are reweighted and combined to produce more accurate classifiers. At each step of the iteration, training samples are reweighted such that incorrectly classified objects get larger weights. Data samples on the margin thus affect the classification rule in each step. This is reminiscent of SVMs which maximize the margin between classes [1]. The connection to SVMs was described in [2] and [3]. Here we adopt the LogitBoost algorithm described in [4]. We use linear classifiers, which are less computationally intensive. Also, the method is parameter-free. The effectiveness of combining linear classifiers using bagging, boosting and random sub-space method was described in [3].

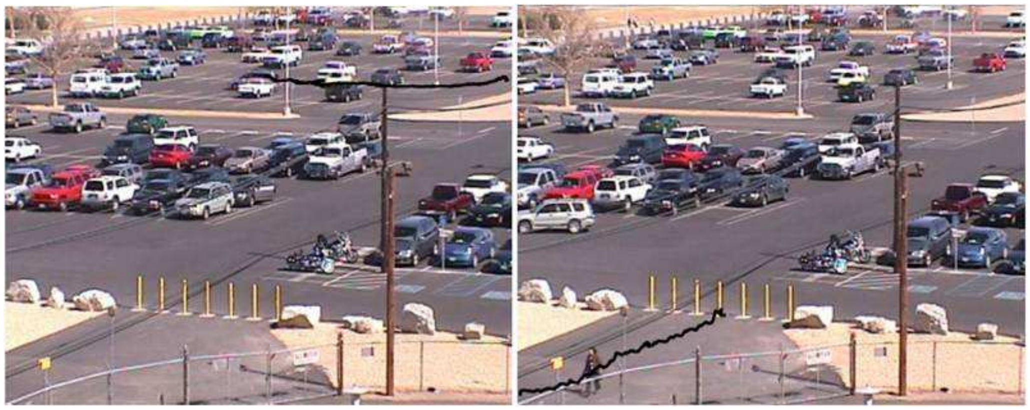

Figure 5: China Lake outdoor surveillance dataset - sample trajectories for vehicle and person.

2.2.3 Behavior Modeling Results

We demonstrate the usefulness of the proposed method for long term behavior recognition and behavior prediction using the China Lake dataset. The China Lake dataset (Figure 5) consists of outdoor surveillance video sequences recorded at the China Lake facility. It consists of continuously-recorded video over a 2 hour period, at a frame rate of 20 fps, on a weekday during which movement of humans and vehicles are observed. Because of the depth of the field of view, objects persist in the scene for several seconds during which they undergo considerable scale changes. The data was recorded on a windy day that caused severe camera jitter. The windy condition proved to be a challenge to the motion stabilization software that was incorporated with the Sony camera. These challenges lead to several false alarms during detection and other low-level vision processes.

Class	Recog.%	Misclassification%
Vehicle is parked	82	12
Vehicle exits lot	93	18
Person enters facility	91	4
Person exits facility	86	8

	C1	C2	C3	C4
C1	73	19	8	0
C2	0	100	0	0
C3	4	0	87	9
C4	0	0	33	67

Figure 6: China Lake behavior analysis recognition rates and confusion matrix.

The dataset was divided into two halves; the first half was used for training and the second half for testing. The training set consists of the following classes of behavior: a vehicle is parked in the lot, a vehicle exits the lot, a person (or a group of persons) exits a vehicle and enters the facility, a person (or a group of persons) exits the facility and walks outside the field of view or enters a car in the lot. There is a significant amount of inter-class variability in these activities because of differences in entry and exit points and rate of execution of activities. The recognition rates (i.e., top match is correct match) and mis-classification rates (i.e., top match is incorrect) are summarized in the first table in Figure 6. The results can be improved by incorporating shape or appearance based recognition, instead of relying solely on trajectories. The ability of the proposed method to predict the future behavior of objects entering the scene is demonstrated using the following target behavior: (C1) a person enters the facility after appearing on scene (i.e., after alighting from a parked car, or entering the scene on foot), (C2) a person appearing on scene walks past the facility, (C3) a vehicle appearing on scene is parked and (C4) a vehicle appearing on scene circles the lot and exits. In all these cases, the predicting classifiers is trained using the initial portion of the trajectories unlike during long term behavior recognition in which the entire trajectories of activities are used in training. Given the first quarter of a test trajectory, the probability that the tracked object will exhibit one of the target behaviors is computed. The resulting confusion matrix is also shown in Figure 6. Rows represent the initial portions of test trajectories and columns represent trained target behavior.

3 Hardware System Architecture

The ONR-funded C2CS research effort carried by the SIG team has focused on the development of a full probabilistic framework and algorithms for object detection & tracking, anomalous behavior detection and sensor integration. Although not directly part of this effort, SIG has also separately developed a hardware system comprised of high-resolution color CCD cameras and a processing system based on a Firewire data network and laptop computers. While this system was not developed as a part of the C2CS effort, it has been useful in allowing the research team to develop, profile and test the algorithms developed for the C2CS effort. The different configurations of the hardware sensor and processing system used in this development effort are describe below, as well as a number of the different system parameters that were optimized to support data collection and analysis for the C2CS effort.

Using this approach, the SIG research team was able to leverage the ONR-funded algorithm development effort to help design a more complete system consisting of algorithms, sensors and processing hardware. As the algorithms matured, and experimentation was performed, both the software and the hardware evolved. We briefly describe the hardware used, and the process of arriving at the current configuration. It should be noted that this parallel development of the C2CS anomalous behavior detection system and the separate hardware processing system also allowed us to demonstrate the real-time capabilities and performance of the SIG research team’s work for the C2CS community at the ONR C2CS PI gathering in May 2008.

3.1 Camera & network considerations

The original sensor hardware was initially chosen to be the IQeye 752 cameras from IQinvision. The IQeye 752 is a progressive scan network (100 Mbps Ethernet) camera with a 1/2 inch CMOS sensor. It can capture 2.0 MP images at a maximum frame rate of 20 fps.

During testing, these cameras exhibited several performance issues. Most serious was the frequent occurrence of dropped frames. These frame drops were often severe, and tended to coincide with significant changes in the image, such as large objects coming into view or fast lighting changes. Typically, between 1 and 5 seconds of video, or 10 to 50 frames of video, were lost at a time. This was enough time for a person to walk about halfway across our field of view or for a car to cross our field of view entirely, making data association significantly more challenging than necessary.

In addition, although the cameras are designed to capture 2.0 MP images at up to 20 fps, we observed a more typical daytime frame rate of 10 to 12 fps. Furthermore, this frame rate was inconsistent over even short periods of time (5 to 10 minutes) and varied with lighting conditions, the amount of activity within the field of view, and image entropy. Although these inconsistent frame rates did not significantly impair our tracking algorithm, they posed an obstacle to anomalous behavior detection. Inconsistent frame rates coupled with a lack of time stamps on the individual images means that reliable object velocity estimates could not be obtained for the anomalous behavior detection algorithm.

Further, the cameras are unable to deliver uncompressed data. The cameras deliver only JPEG compressed data which, even at the highest quality setting, introduces block artifacts into the images. JPEG compression works on 8x8 blocks of the image causing the value of any one pixel in a given 8x8 block to depend on the value of every other pixel in that block. Practically, this means that when an object enters an 8x8 block and changes the color values of 2 or 3 pixels in the raw image, it changes the color values of all 64 pixels in the JPEG compressed image. This significantly complicates color modeling by introducing a strong, unnecessary spatial dependency between pixels. The easiest way to mitigate this problem is to down-sample the image by a factor of 8 in both dimensions, removing the spatial correlations in the data. However, this negates the advantage of using a high resolution sensor and wastes unnecessary CPU cycles.

Finally, JPEG decompression became the significant bottleneck in the performance of our tracking algorithm. Although we achieved a frame rate of 7 to 8 fps on a live image stream on our target architecture, we determined that further speed increases could only be achieved by reducing the time spent on image decompression. The easiest way to address this problem was to move to cameras that deliver raw image data.

After encountering several performance issues with the IQinvision cameras, we chose to replace them with DBK 21AF04 cameras from The Imaging Source. The DBK 21AF04 is a progressive scan Firewire (IEEE 1394) camera with a 1/4 inch CCD sensor. It delivers 640 by 480 pixel uncompressed Bayer pattern images at stable frame rates up to 60 fps.

The switch to The Imaging Source cameras has removed all of the performance issues described above for the IQinvision cameras. However, due to the Firewire interface, and the limitation on the cable length for Firewire communications, the system cannot position the cameras as far from the laptop computer as was possible with the Ethernet cameras.

We have experimented with a variety of CS-mount lenses. Lens selection is a trade-off. Shorter focal lengths provide a larger field of view and greater depth of field but fewer pixels on target and greater image distortion. Our tracking and anomalous behavior detection algorithms are insensitive to image distortion, and our targets have been relatively close to our cameras. So we have chosen to use fixed length 3.6mm Computar lenses. This is a very short focal length that provides a wide field of view.

3.2 Computational platform

We initially planned to implement our tracking and anomalous behavior detection algorithms on a DSP (digital signal processor) to facilitate real-time performance. However, after porting large portions of the tracking algorithm from our development environment in MATLAB to the compiled language C, it became apparent that we could achieve real-time performance on a typical Microsoft Windows XP laptop computer without the need for specialized hardware. This change of a target processing platform allowed more time on algorithm development and less on implementation. Using a laptop instead of a DSP also means that the implementation is more portable. Our implementation can now be run on any Microsoft Windows XP PC with the .NET Framework and hardware comparable or superior to our target hardware.

We chose a Dell Latitude D620 with an Intel Core 2 T7200 clocked at 2.00GHz with 2.49 GB of RAM as our target hardware. We connect the camera to the laptop through a IEEE 1394 CardBus interface that also powers the camera.

We chose ANSI C as the development language for the tracking and anomalous behavior detection algorithms. This means that the implementation of our tracking and anomalous behavior detection algorithms is easily portable to any operating system and architecture with an ANSI C compatible compiler. We chose Microsoft Visual Studio C++ and the .NET Framework as the development platform for our visualization and GUI. We used the IC Imaging Control libraries for Visual C++ to acquire images from the cameras.

4 Sensor management

4.1 Mechanical pan/tilt/zoom

The multi-senosr system was originally conceived with the cameras attached to a mechanical pan/tilt servo, with automated zoom capability. This system was intended to provide a greater effective range for the field of view, while also providing high resolution images for the software system. However, during the course of the development, it became clear that this approach had several potential problems.

One of these problems was the speed of transition from one state to another. Existing systems are often slow, taking on the order of a second to achieve a new state, and settle down. During this process, the images obtained would experience significant blurring due to the motion of the camera, and would be of very limited use to the tracking software. Also, these systems are not extremely precise, and the actual motion achieved (and reported) would be merely an approximation of the desired motion.

Another factor was the cost of missed opportunity. While the system is looking at a certain section of the observable scene, there is a significant portion of the image which is going unobserved. Moreover, when the system is in motion, there is effectively no observation of the world, and the cost of missed opportunities jumps even more.

Finally, the price of a PTZ system is significant, relative to the price of the camera, making it effectively cheaper to use three separate cameras to cover a wider area rather than one mobile one. Alternatively, a single high resolution, wide angle camera can be purchased less expensively, and producing more information with greater persistence of surveillance.

Given these significant drawbacks, and the ease and expensive of alternative cameras, the mechanical pan/tilt/zoom capability was discarded.

4.2 Virtual pan/tilt/zoom

As an alternative approach to the mechanical system, a virtual pan/tilt/zoom capability was proposed. This would use a wider angle view on a superior camera, and provide the same capability as the mechanical system. The benefits of the system include reduced price, persistent observation of the scene, and no motion blur. Additionally, the system has no moving parts, making it more reliable and less prone to physical failure, while consuming less power.

The motivation behind this virtual PTZ is to focus the tracking on the areas of interest in the scene. Simple heuristic methods can be used to watch for significant change in the wide field of view, to look for new objects. Once an object has been detected, the tracking algorithm can focus on the regions of the image relevant to following this object. Additionally, processing can be performed at various levels of resolution in the image, depending on the need of the tracker, providing the same benefits of a zoom capability.

Using this focus of interest, the analyst could then be presented solely with the 'virtually zoomed' area relevant to the detected anomalies. Alternatively, the system can present the entire field of view, with graphical enhancements to draw the analyst's attention to the areas of interest, which are displayed in the highest resolution available. It is this latter scenario that has been built into the multi-sensor software application. This framework essentially allows us to decouple the resolution choices used for the processing of the foreground and background portions of the pixels data, improving the system computational performance while still supporting extraction and visualization of the regions of interest.

4.3 Data compression

It is interesting to note that several transition applications for the multi-sensor project often involve a limited data transmission capability from the sensor to the analyst. As currently designed, many of these systems will send the data out either at a significantly lower frame rate, or in a significantly compressed form, resulting in a drastic loss of relevant information.

The development of the virtual PTZ system has created the potential for an improved solution to this problem. Rather than present cropped, compressed, or low frame rate data to a user, the system can present entire field of view, with more information contained specifically in the areas that are of interest to the analyst. These pixels relating to the foreground objects, particularly those deemed anomalous, can be transmitted with full color data at maximal resolution, resulting in no loss of information for the analyst. The background can be presented in a compressed manner, in one of several ways.

One of these approaches would involve no data sent from the sensor for those pixels relating to the background. Instead, the user relies on a canonical model of the background, such as a single frame sent at the start of the transmission. This template for the background would have the foreground pixels overlaid on top of it, to provide contextual information to the analyst.

An alternative approach would continue to send information on the background pixels, but this data would be transmitted as significantly lower resolution, using only grayscale information. Additionally, the background image could use a reduced palette, requiring fewer bits to represent the range of grayscale intensities. The visual effects of this type of compression can be seen in the output images in the application software.

The final alternative presented would maintain continual updates to the background at full resolution and with the full color palette available. However, the effective frame rate for the background would run significantly slower than the full speed provided for the foreground. Instead, the background would be updated progressively across scan lines at each time step, utilizing whatever available bandwidth is left over after transmitting the foreground. Each time step would pick up where the last one left off, resulting in a continual, cyclic update to the areas of least interest to the analyst, adaptively changing the update rate to accommodate the available transmission capability at the time.

Initial results from using this type of intelligent compression scheme, utilizing the reduced resolution scheme for the background pixels, indicate that approximately 40 times fewer bits are required to transmit the data, when compared to the raw data files, without a loss of information in the relevant areas of the image. With the additional use of existing lossless image compression techniques, this reduction can achieve an average of 110 times the original image size. Compare this to a typical jpg compression format, which causes significant loss of data, particularly in the fine details, which can achieve approximately 15-20 times fewer bits for the data observed, depending on the 'quality' of the compression used. This allows for significantly increased data throughput to the analyst, while still providing nearly all of the relevant information that the analyst will need.

5 Compressive sampling

The SIG C2CS effort at object tracking as seen in the previous sections has been shown to provide an effective approach for data compression if only moving foreground objects are of interest in a video scene. This would be appropriate for applications where the goal is to detect & track objects as well as to detect anomalous object behaviors. In such application, we could clearly compress high rate data streams based on selective representation of objects and background to send only data relevant to tracking objects.

The concept of compressive sensing (CS) also allows information to be extracted from sensor data (including video image sequences) using significantly fewer samples relative to conventional uniform sampling techniques. As indicated in our Year III planning letter, this related technology has been the subject of an effort to examine the approach of combining CS approaches to track objects in time-multiplexed video imagery.

5.1 Background on Compressive Sensing

Compressive Sensing is the theory which allows one to sample significantly less observations than the Shannon sampling theorem would normally permit. For a band-limited signal, with highest possible frequency = f_M, Shannon theorem states we need to sample at a rate of at least $2*f_M$ for perfect signal reconstruction. The compressive sensing theorems of Candes and Tao [5], and Donoho [6] have shown that, with modest additional assumptions, one can have perfect signal reconstruction with far fewer samples than the Shannon Theorem implies. These assumptions are:

1. There is structure to the signal (i.e. the signal is not random). $\leftrightarrow$ There exist a basis for which the representation of the signal of interest is sparse $\leftrightarrow$ The signal is compressible
2. The notion of "observing a sample" is generalized to include linear projections of the signal in addition to the instantaneous signal amplitude.

Assumption 1 is easily satisfied by almost all relevant signals one wishes to exploit, while assumption 2 is satisfied by assuming the existence of an analog device which can perform these projections during the sampling process.

Let us assume that the underlying, high resolution, signal is given by **F**. The basis assumed under assumption 1, is represented by the matrix **B**. Then assumption 1 says that **F** = **Bα**, for some coefficient vector **α**. The projection samples in assumption 2 can be represented by a "sampling" matrix, **P**. Each row of **P** represents a sampling of the underlying signal. The dimensions of **P** are MxN, where M<<N. This means we are only taking M samples of the signal information, but previous theory would imply the need for N samples to be taken. The compressed sensing equation becomes.

$$\mathbf{Y} = \mathbf{PB}\boldsymbol{\alpha} \tag{16}$$

where the observed signal samples are given in the M dimensional vector **Y**. To recreate the desired signal, one needs to estimate the coefficient vector **α**. It has been shown that assumption 1 allows us to estimate the coefficients required for perfect signal reconstruction through the following optimization:

$$\boldsymbol{\alpha}_{\mathbf{optimal}} = \underset{\boldsymbol{\alpha}}{\mathbf{argmin}} \left\| \boldsymbol{\alpha} \right\|_{\mathbf{1}} \quad \textbf{with } \mathbf{Y} = \mathbf{PB}\boldsymbol{\alpha} \tag{17}$$

There are numerous threads of research regarding the solution to equation 2. Among many techniques, greedy algorithms and Bayesian algorithms have been developed to estimate the appropriate coefficient vector alpha; each of which has its own strengths. In the subsequent analysis, the Bayesian approach developed by Ji, Xue, and Carin [7] is utilized. They have chosen a noisy model for the underlying measurement vector, so the compressive sensing equations become

$$\mathbf{Y} = \mathbf{PB}\boldsymbol{\alpha} + \mathbf{n} \tag{18}$$

Where **n** is a noise vector which represents measurement noise as well as coefficients which are not identically 0 but can effectively be treated as zero. The Bayesian approach to the reconstruction then becomes

$$\hat{\alpha} = \mathbf{argmin}_{\mathbf{v}} \left\| \mathbf{Y} - \mathbf{PB}\boldsymbol{\alpha} \right\|_2^2 + \boldsymbol{\rho}\sum_{\mathbf{n=1}}^{\mathbf{N}} \left| \alpha_{\mathbf{n}} \right| \tag{19}$$

There has also been renewed interest in looking at the relationship between the projection and basis matrices (P and B), as there are key theoretical subtleties involved with the design of these matrices which allow for perfect signal reconstruction. There is also a thread of research involving the noise properties of the measurements observed and the effect on algorithms for approximating the solution to Equation 19. A brief tutorial paper by Baraniuk [8] is a helpful start to those who wish to review many of the foundations of the theory and the author also maintains a website of useful references for those interested in the latest research (http://www.dsp.ece.rice.edu/cs/#app).

5.1.1 Compressive Imaging

The underlying theory with regards to imaging sensors is the same as that for signals. The major difference is that imaging sensors are not generally conducive to applying the theory to real world applications. For example, an imaging sensor designed for surveillance applications would currently collect video image information for on-board or ground-based exploitation processing. The Focal Plane Array (FPA) is the typical device for collecting the image information and these are designed and built to collect the required number of pixel intensities to represent the scene being surveyed. A design involving compressive imaging would have to provide an optical architecture for the standard FPA to collect projections of the scene or offer a fundamentally different type of FPA. In order to fundamentally improve current imaging systems, one would want an architecture which will collect the same number of measurements as the current imager, but is able to use the compressive imaging theory to fundamentally improve some aspect of the scene which is being sensed. This is a subtle but important point: if the current application solution employs a FPA which has **K** elements and samples a **K** dimensional image, then for the compressive imaging theory to be of practical utility, one would want to measure **K** pieces of information and rebuild an image scene with **N**>>**K** dimensions. For medical imaging applications, the desired image to be sensed was always **N** dimensional with a **K** dimensional sensor. The fundamental question is: How can the current image be improved from what the current sensor produces by employing compressive imaging? Some possible applications and architectures are presented next.

5.1.2 Applications

In DoD surveillance applications, the Field-of-Regard (FOR) has become too large for a single imaging sensor to capture the activities for timely exploitation. Using a scanning imaging system will require large areas of the FOR to go un-sensed while the scanner is imaging another area. Multiple sensors or platforms will result in a possibly unmanageable data glut and require extra processing (such as sensor registration or mosaicing). This application has been addressed by Muise and Mahalanobis [9] and Muise [10]. For image-based Automatic Target Recognition (ATR) applications one has the distinct possibility that there is significant mismatch between what is being sensed, and what is optimal for the ATR to make a decision. For example, the imaging sensor is perhaps wasting photon collection resources to image irrelevant and confusing clutter information when more photon collection on the target of interest could increase the ATR performance. This application has been presented by Mahalanobis and Muise [11]. In object tracking applications, the current estimate of the object state is only as good as the revisit rate of the imaging sensor. One would ideally want to use the imaging sensor to help determine the current position of the target at time samples determined by the object's velocity; while imaging other, static, portions of the scene at a lower sampling rate.

If one was to use standard imaging devices, each of these applications describes the desire to exploit an image with more information than what is actually being collected. Thus, is it possible to get this extra image information from fewer sensed pixels? We next address a different paradigm in extracting intensity information from a detector array which will allow one to estimate what is happening "between the frames" of a video sequence.

5.2 Time Multiplexing

We set up the problem and application as a sampling problem for video imagery. We also wish to establish applications for which traditional video sampling (1 image frame per time interval) will not capture the dynamic nature of the underlying scene.

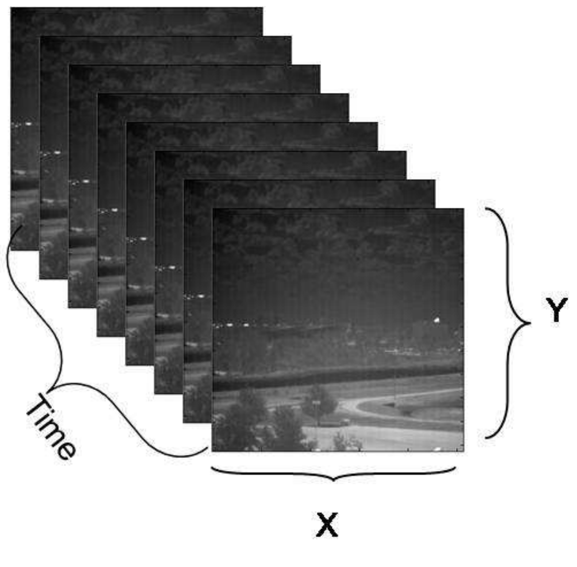

Figure 7: A Representation of an analog cube of image intensity information with spatial and time axes.

In Figure 7, we show a sequence of image frames which represent a video sequence. It is shown as a discrete representation of multiple frames of standard imagery. In order to characterize a time multiplexing imager, we need to consider this data in the analog sense. (i.e. a dynamic scene represented continuously in time). The sampled representation implied by this figure is for illustration, we will not be sampling this continuous stream of intensity values in the traditional way.

Let us consider that for our application, a particular signal-to-noise ratio (SNR) is required for the exploitation algorithms to have adequate performance. This required SNR will lead directly to a required integration time, $\mathbf{T_s}$ for the detector elements in the focal plane array (FPA) to gather sufficient photons to meet the SNR constraints. For typical video cameras fielded to address typical exploitation tasks, these constraints do not prohibit us from capturing the essential dynamic of a typical scene for timely exploitation. For most well-designed camera and for most dynamic scenes, standard 30 or 60 Hz frame rates seem to be adequate to capture imagery at suitable SNR levels for exploitation purposes.

Let us then consider applications for which standard sampling of the data cube of interest does not result in exploitable video sequences. These applications involve the attempt to detect and track objects which are moving much faster than the required integration time, $\mathbf{T_s}$, will be able to capture. If there is significant motion during the integration time, $\mathbf{T_s}$, then traditional image sampling will not capture this motion.

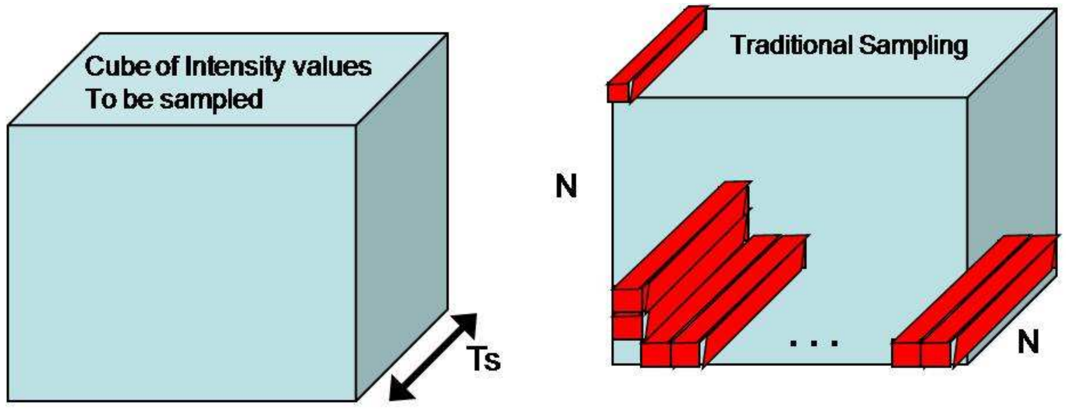

Figure 8: Left - The cube of image information which needs to be sampled. The integration time Ts reflects that all of the photons within this cube are required for sufficient SNR imagery. Right - The traditional methodology for sampling a video frame. We integrate the photons along the red rectangular elements to arrive at a pixel intensity value.

Consider the surveillance application to detect and track moving objects where the sensor properties and/or target dynamics lead to poor SNR imagery. This could be manifested in the need for utilizing and uncooled IR camera for cheap and/or small platform night applications. The poorer SNR properties of an uncooled IR camera could be solved by a longer integration time. However, this increased integration time could lead to motion blur for fast moving targets. Even typical 30 Hz video can lead to significant motion for highway traffic between frames. An increase of the integration time to accommodate a poorer performing sensor may lead to the inability to sense highway traffic. Poor lighting conditions in general would lead to the desire for a longer integration time and affect performance similarly. Turning attention away from camera constraints and toward the scene dynamics, any high velocity target will be difficult to track with standard video frame rates. Tracking bullets or missiles can often require specialized high frame rate cameras to capture and track the object motion. As the frame rate increases, the integration time decreases and the image SNR will become poorer. There are only so many photons to collect during $\mathbf{T_s}$.

We propose to extract data from the sensor at the same essential rate as with traditional sampling, but with information being extracted throughout the entire integration time $\mathbf{T_s}$. Consider Figure 9 which describes an alternate methodology for sampling the data cube.

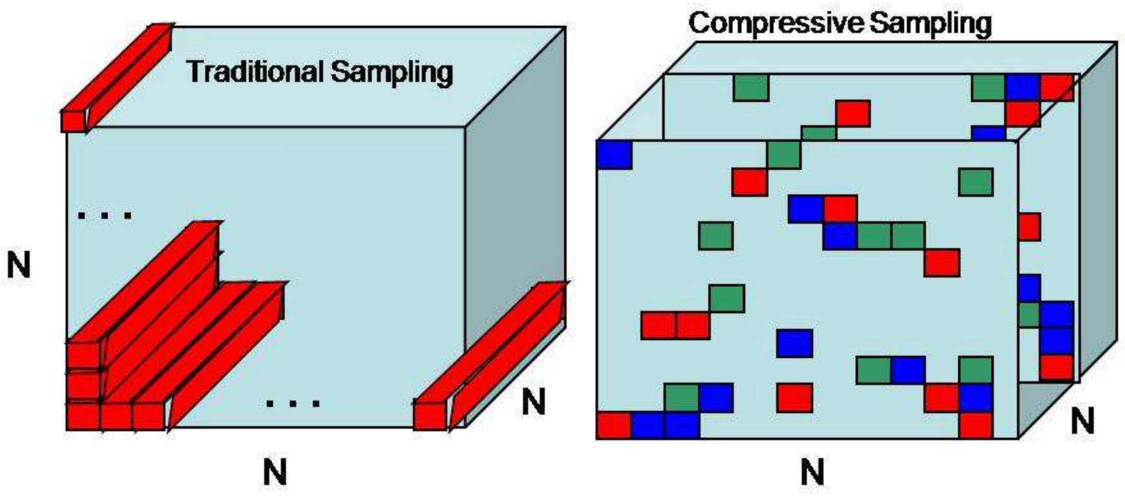

Figure 9: Left – The traditional methodology for sampling a video frame. We integrate the photons along the red rectangular elements to arrive at a pixel intensity value. Right – A compressive sampling architecture. The collections of all same colored "pixels" are integrated together as a large "super-Pixel". There are N super-Pixels collected at N different time intervals. The spatial distribution of the same colored pixels are randomized.

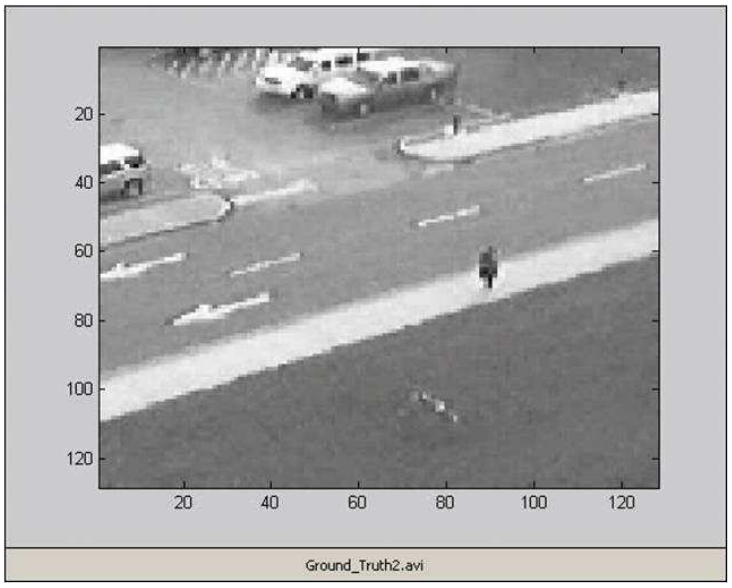

Figure 10: A 128x128x128 data cube used for simulations

There exist smart and adaptive FPA technologies which can gather this encoded information in the manner outlined in Figure 9. In order to test the ability for the compressive measurements to "see" between samples of a traditional camera we use some standard video data collected and test the concepts by simulation. The video sequence utilized is presented in Figure 10.

This video is taken as the ground truth high frame rate sequence which we attempt to reconstruct. It is 128x128 units of spatial dimension and 128 time units. We assume that the required integration time for acceptable SNR imagery is the entire 128 time samples or $\mathbf{T_s}$ = 128. This does not exactly conform to our specified applications where the motion is faster than the required SNR will support, but the video will serve as a proof-of-concept simulation. The results of sampling this data cube by the methodologies in Figure 9 are shown in Figure 11.

The mathematical formulation for the encoding and the compressive decoding is given in the next section.

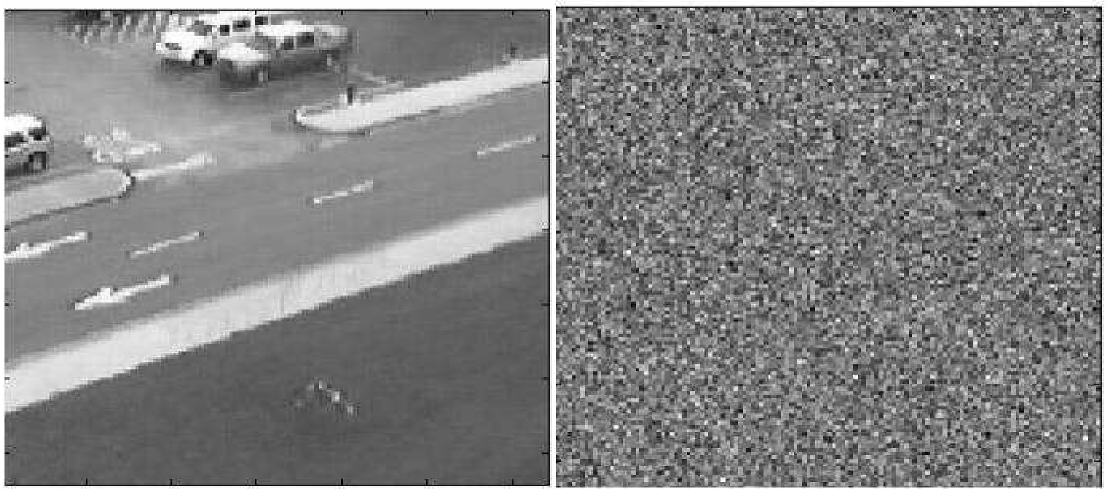

Figure 11: Left – The traditional methodology for sampling the data cube. The moving object has been averaged away during the integration at the FPA. Right – The result of compressive sampling on the data cube. There are 128x128 numbers (the same as the traditional sampling). The information appears random, but it is encoded to allow for estimation of the moving object during the sampling interval.

5.3 Mathematical Foundations

Let **J** be the NxNxN cube of image intensity information which one would integrate into one video frame. Further, let $\mathbf{J}_i$ be the ith time slice of this video cube. Further, let us reorder $\mathbf{J}_i$ into a vector by lexigraphical ordering and without ambiguity, we also refer to this collection with the variable **J**.

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & \mathbf{J}_2 & \cdots & \mathbf{J}_N \\ \vdots & \vdots & & \vdots \\ & & & \end{bmatrix} \qquad (20)$$

Thus, each column of **J** represents one time slice of the data cube, and **J** is an N^2xN matrix of intensity values. With this notation, the image collected on the FPA for a traditional video camera would be

$$\mathbf{I}_T = \mathbf{J} \begin{bmatrix} 1/N \\ 1/N \\ \vdots \\ 1/N \end{bmatrix} \qquad (21)$$

This is represented as the image presented in the left side of Figure 11. In order to describe the collection of compressive measurements, let us define some intermediate variables:

Let $\mathbf{S}_i$ = the N^2x1 vector which is 1 for a pixel element belonging to "super-Pixel" i. Thus, since there are N defined super-Pixels, there are N vectors $\mathbf{S}_i$. We will also be defining these superPixels at each of N time steps throughout the data cube, thus let

$$\mathbf{S}_\mathbf{i}^\mathbf{j} = \mathbf{S}_\mathbf{i} \;\; \textit{for the } j^{th} \textit{ time step} \tag{22}$$

Further, let

$$\mathbf{S}^\mathbf{j} = \begin{bmatrix} \mathbf{S}_1^\mathbf{j} & \mathbf{S}_2^\mathbf{j} & \cdots & \mathbf{S}_\mathbf{N}^\mathbf{j} \end{bmatrix}^\mathbf{T}; \tag{23}$$

an NxN^2 matrix. Then for the jth time step, we collect the information

$$\mathbf{P}_\mathbf{j} = \mathbf{S}^\mathbf{j}\mathbf{J}_\mathbf{j}; \tag{24}$$

which is an Nx1 vector of projection information. The collection of this information over all time steps yields the collected information

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_1 & \mathbf{P}_2 & \cdots & \mathbf{P}_\mathbf{N} \end{bmatrix} \tag{25}$$

This is N^2 numbers and is represented by the array given in the right side of Figure 11.

In order to decode this information with the theory of compressive sensing, we must relate the data we wish to rebuild to a sparse representation. Clearly the difference image between two successive time steps will be sparse as the static portions of the scene will be zero. Consider the average image over a considerable time span, $\bar{J}$. Then the encoded average image will be given by

$$\begin{aligned}\overline{\mathbf{P}} &= \begin{bmatrix} \mathbf{S}^1\overline{\mathbf{J}} & \mathbf{S}^2\overline{\mathbf{J}} & \cdots & \mathbf{S}^\mathbf{N}\overline{\mathbf{J}} \end{bmatrix} = \begin{bmatrix} \mathbf{S}^1 & \mathbf{S}^2 & \cdots & \mathbf{S}^\mathbf{N} \end{bmatrix} \begin{bmatrix} \overline{\mathbf{J}} & \mathbf{0} & & \\ \mathbf{0} & \overline{\mathbf{J}} & & \\ & & \ddots & \mathbf{0} \\ & & \mathbf{0} & \overline{\mathbf{J}} \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{S}^1 & \mathbf{S}^2 & \cdots & \mathbf{S}^\mathbf{N} \end{bmatrix}\overline{\mathbf{J}} \;\; \textit{for a compact representation}\end{aligned} \tag{26}$$

We form the difference codes by

$$\mathbf{D} = \mathbf{P} - \overline{\mathbf{P}} = \mathbf{S}\left(\mathbf{J} - \overline{\mathbf{J}}\right) \tag{27}$$

Only those pixels which have motion during the sequence will contribute energy to the encoded data in the matrix **D**. This is the essence of the sparse data collected. The next step is to decode this information into an estimate to which pixels in the *k*th time step have exhibited motion.

5.3.1 Finding Pixel Motion

Consider the kth time step. We select the k-1, k, and k+1 columns of the matrix **D** and call this collection of data $\mathbf{D_k}$. These columns contain the 3xN pieces of information gathered from the *k-1*, *k*, and *k+1st* frames of imagery from the data cube. We further assume that the image $\mathbf{J_k}$ has not changed during this, very short, collection. Thus, our collection of information is given by the simple linear equation

$$\mathbf{D}_k = \begin{bmatrix}\mathbf{S^{k-1}} & \mathbf{S^k} & \mathbf{S^{k+1}}\end{bmatrix}\left(\mathbf{J_k} - \overline{\mathbf{J}}\right) \tag{28}$$

This is an Nx3 collection of encoded numbers. If we define a basis, **B**, for which our difference image resides, then we have the system

$$\mathbf{D}_k = \begin{bmatrix}\mathbf{S^{k-1}} & \mathbf{S^k} & \mathbf{S^{k+1}}\end{bmatrix}\mathbf{B\alpha} \quad where\ \mathbf{B\alpha} = \left(\mathbf{J_k} - \overline{\mathbf{J}}\right),$$

Or simply

$$\mathbf{D}_k = \mathbf{SB\alpha} \tag{29}$$

This is an underdetermined linear system where $\mathbf{D_k}$ is a 3Nx1 observation vector, **S** is an 3NxN^2 code matrix, **B** is an N^{2}xN^2 basis function matrix, and $\boldsymbol{\alpha}$ is an N^{2}x1 coefficient vector. This is exactly the sparse representation-compressive sensing setup where the solution is given by Equation $\hat{\alpha} = \mathbf{argmin_v} \left\|\mathbf{Y} - \mathbf{PB\alpha}\right\|_2^2 + \boldsymbol{\rho}\sum_{\mathbf{n=1}}^{\mathbf{N}}\left|\alpha_{\mathbf{n}}\right|$ (19) as

$$\hat{\boldsymbol{\alpha}} = \mathbf{argmin_v} \left\|\mathbf{D_k} - \mathbf{SB\alpha}\right\|_2^2 + \boldsymbol{\rho}\sum_{\mathbf{n=1}}^{\mathbf{N^2}}\left|\boldsymbol{\alpha}_{\mathbf{n}}\right| \tag{30}$$

This minimization is solved by a Bayesian algorithm developed by Ji, Xue, and Carin [7]. The results on the data cube from Figure 11 are given in Figure 12. These results utilize a basis set which consists of shifted box functions. Each basis function is a 3x3x3 collection of ones and this basis set is shifted over all the N^2 pixels in the underlying difference image being estimated.

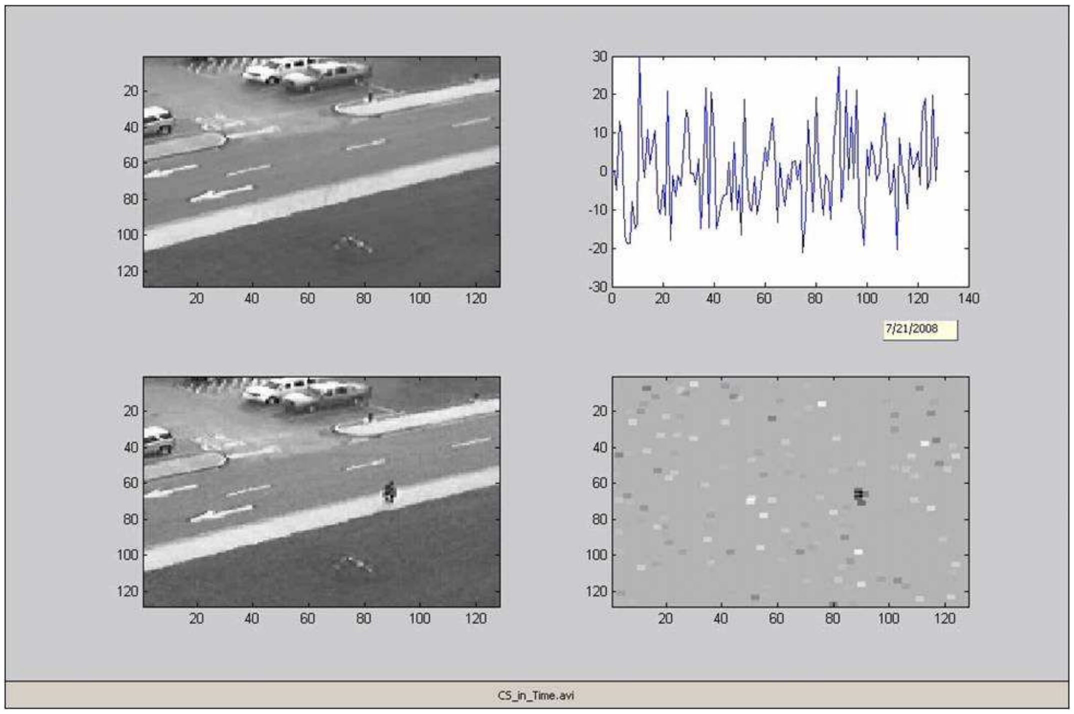

Figure 12: Top Left: The information scene by a traditional camera. Top Right: the vector of data collected during the compressive sampling (a particular column of figure 4). Bottom left: The ground truth video. This is unobservable to the system as we are assuming the motion is too fast from traditional sampling. Bottom right: The successive estimates of the difference imagery collected and estimated as per equations 28 through 30.

5.4 Implementation

Given the algorithm outlined above, one must design an implementation strategy that allows computation as the streaming encoded information is output from the FPA. Firstly, $\overline{\mathbf{P}}$, the average encoded information vector is collected and updated as new data is collected. This process should have a relatively long time constant as we are trying to capture the static portions of the scene. However, there should be careful attention to not over average this information as slow environmental and lighting changes should be reflected in the current estimate of $\overline{\mathbf{P}}$. Then, after 3 frames of encoded information have been collected, we have the information required to populate the information required to formulate Equations 28 through 30 and the estimated motion is calculated. The new information is added to the columns of the encoding matrix S and the data vector D and Equation 30 is solved once again. A further coupling between subsequent estimates can be made by increasing the dimensions of the basis function to 3x3x5 pixel box functions. This was seen to be beneficial in reducing some of the noise.

A typical camera would integrate over time (in this case 128 time steps) and produce the video such as Figure 11; when compressively sampled, it would look more like the right side of Figure 11 . Using the algorithm from equations 28 through 30, we estimate the difference image between successive frames. With this data, we can track groups of moving pixels. Once these tracks are identified, then we know the location of moving object between frames. Shifting the collected data spatially will lead to the ability to reconstruct the image of the moving object and place it in the average scene in the correct location in space and time. Our initial simulations show that this approach clearly can find and track moving objects.

Our initial implementation, however, does give a poor result for the appearance of the moving object as the object itself is smoothed over space and time – i.e. the object image is smeared. A better approach would be to find a more accurate reconstruction for the moving object based on hypothesis about specific pixels that are part of the object based on the difference image. These potential improvements will the subject of future work extending these initial results.

In the last section, we turn to exploitation. In the context of general compressive imaging, can we perform object classification directly on the gathered compressive measurements without first reconstruction the underlying image?

5.5 Discrimination using Compressive Measurements

In this section, we examine the ability to discriminate objects by directly operating on the compressive measurements without first reconstructing the image. As we have shown earlier, images of specific objects of interest can be reconstructed by minimizing a weighted L2 norm metric. This results in a closed form analytical solution for the reconstruction matrix **R** which related the reconstructed image **y** to the measured projections **u** as

$$\mathbf{y} = \mathbf{Ru}. \qquad (31)$$

Let us now consider the quadratic correlation filtering detection algorithm trained to detect patterns in the images. The image **y** is transformed by the QCF matrix **S** to a set of features **v** that represent energy in clutter and target basis, i.e.

$$\mathbf{v} = \mathbf{Sy} \qquad (32)$$

However, since **y** is estimated by Eq. 15, it is easy to see that the compressive measurements can be directly related to the QCF features as

$$\mathbf{v} = \mathbf{SR}\,\mathbf{u} \qquad (33)$$

The advantage of Eq. 31 is that the dimension of the matrix **SR** is considerably smaller, and therefore few computations are required to directly calculate **v** from **u**, rather than first reconstruct the image via Eq. 33, and then applying Eq. 32.

5.6 Preliminary Results

Eq. 33 shows an approach for directly embedding the QCF discrimination algorithm (trained on high dimensional images) directly into the compressive sensing process. This allows the compressive measurements to be directly used for exploitation without the need to first reconstruct the image. However, the question arises how this compares to a discrimination algorithm which is trained directly the compressive measurements (i.e. not only the high-dimensional images, but in the considerably smaller compressive measurement space).

Preliminary results show that QCF trained on the images (in the high-dimensional space) does not perform as well as when it is trained directly on the smaller dimensional measurement vectors. However, this can be due to several reasons that require further investigation. The data below was obtained by using every other image in the data set to train and to test. The high correlation between the test and training condition may favor the results obtained on the smaller dimensional compressive measurements. Secondly, since the relations are all linear, there can be no fundamental gain in processing and at best, the full dimensional QCF in Eq. 33 will perform as well as when the algorithm is directly trained using the compressive measurements. However, if non-linearities were introduced in the reconstruction matrix **R**, or if the weighting of the norm were to be varied to further support discrimination, there will be a premise where differences (and perhaps improvements) in the results can be expected.

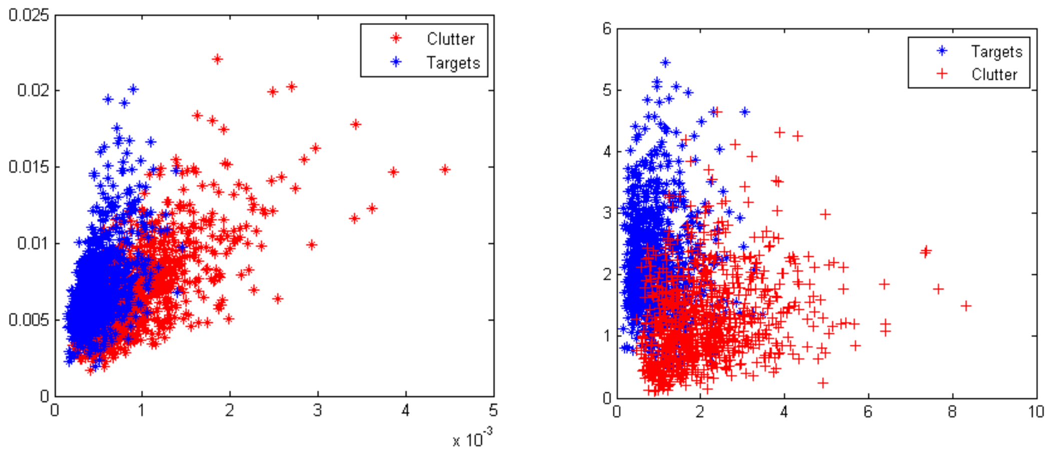

Figure 13: The scatter plots of the QCF features computed by (a) embedding the high-dimensional solution in the reconstruction process does not produce as good a result as in (b) where the data shows the results of training a QCF solution directly on the lower dimensional compressive measurements.

6 Technology Transition

One of the key successes of this SIG C2CS project has been the transition of the technology developed here to other DoD-funded research effort in several different key application areas.

This transition has taken place in two key areas:

(1) Transition of key concepts regarding the integration of a software algorithm and a human analyst through the use of *active learning*, and
(2) Transition of conceptual mathematical models and approaches for Bayesian tracking to airborne persistent surveillance applications.

In the following sections, we provide more details on these transitions and their initial results.

6.1 Airborne Persistent Surveillance

The airborne persistent surveillance application has become a critical area for automatic processing of sensor data to track and detect moving objects as well as to perform behavior analysis of those objects detected. While this application area is beyond the scope of this C2CS project, SIG has achieved transition of significant tracking technology to support persistent surveillance. One key goal for persistent surveillance is to achieve real-time capability for processing sensor data. This capability is critical in order to provide improved response capability to events of interest. One approach to providing this capability using SIG's C2CS developed technology is to deliver high quality metadata of a surveillance scene using tracking, classification metadata derived from raw imagery on the surveillance platform. This approach has the potential to avoid pitfalls of off-the-shelf compression method because it can results of tracking algorithms to identify objects of interest and can separately provide scene data provided for contextual information.

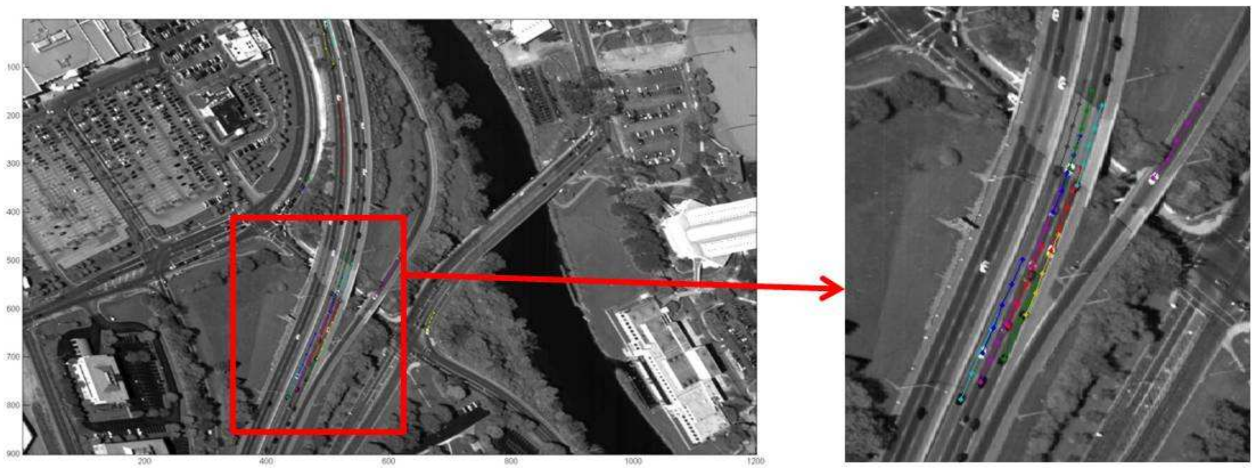

Figure 14: Initial results showing singe frame capture from implementation with fully automated registration with detection and tracking performance comparable to ground-based video for multiple small objects

Based on technology initially developed under ONR C2CS funding, SIG has completed efforts for development & testing detection and tracking algorithms for airborne IR data sets funded through NVESD. As part of this effort, SIG has adapted the original approaches for background modeling, object detection and Bayesian tracking and has created a new implementation to account for the different problem constraints of the wide-area persistent surveillance problem. This application of the C2CS technology has been applied to an Army-funded effort for tracking using Army UAV data sets. Figure 14 shows some initial results where this new adapted tracking approach has been applied to persistent surveillance data. Additionally, SIG has been funded for an Air Force effort applying this extended technology to CLIF persistent surveillance data sets with funding through AFRL.

6.2 Active Learning

In the initial work performance for this C2CS effort, SIG examined the concept of using expected information gain to make decisions about sensor deployment. In particular, this concept was manifested as the sensor management architecture (SMA), where expected performance gains in object tracking are balanced against costs of sensor employment/redeployment. An important extension of this work under C2CS was a transition of this same concept to the use of expected information gain that could be used to improve performance of a human analyst that is tasked to process EO or IR imagery in order to detect targets through change detection.

Under this approach, the use of relative information gain was transitioned to and Army-sponsored (NVESD) program where the additional data was available as labels provided by a human analyst. Figure 15 shows some initial results where this learning concept was applied to a change detection application for finding targets in airborne imagery. Additional work was also done in extending this application of active learning to Army NVESD airborne target detection to both off=-line batch processing as well as real-time streaming video imagery. The development & testing on airborne IR data sets funded through NVESD and additional follow-on work to this effort is being pursued by SIG for a number of different customers.

The initial results for this application of active learning to a combined system with and automated target classification algorithm and an "analyst in-the-loop" demonstrate effectiveness of active learning for improving analyst performance and efficiency. Figure 15 shows some basic performance curves where analyst processing time (in minutes) is reduced at the same time as analyst performance is improved through the reduction in target false-alarm reductions.

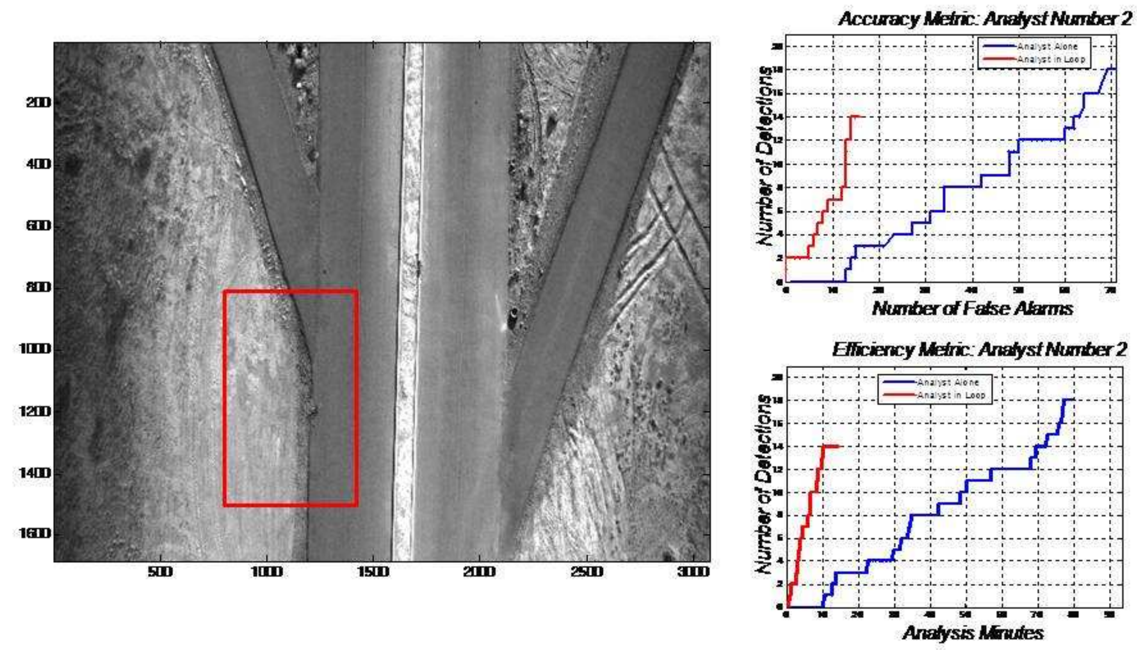

Figure 15: Initial results showing use of active learning for improving analyst performance and efficiency

References

[1] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley, 2001.

[2] R. E. Schapire. *Nonlinear estimation and classification*, chapter The boosting approach to machine learning: An overview. Springer, 2003.

[3] M. Skurichina and R. P. W. Duin. *Pattern Analysis and Applications*, chapter Bagging, boosting and the random sub-space method for linear classifiers. Springer, 2002.

[4] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: A statistical view of boosting. *Annals of Statistics*, 38(2):337–374, April 2000.

[5] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[6] D. L. Donoho, "Compressed sensing," *IEEE Trans. Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[7] S. Ji, Y. Xue and L. Carin, "Bayesian compressive sensing," to appear in *IEEE Trans. Signal Processing* (available at http://www.dsp.ece.rice.edu/CS/BCS_preprint.pdf)

[8] Richard Baraniuk, "Compressive Sensing." (IEEE Signal Processing Magazine, July 2007)

[9] R. Muise and A. Mahalanobis, "Computational sensing algorithms for image reconstruction and the detection of moving objects in multiplexed imaging systems", SPIE Defense and Security Symposium, Proc. SPIE 6977, 69770M, March 2008

[10] R. Muise, "Compressive Imaging: An Application", Submitted to SIAM Journal on Imaging Sciences, April 2008, In review.

[11] A. Mahalanobis, R. Muise, "Object Specific Image Reconstruction using a Compressive Sensing Architecture for application in surveillance systems", submitted to IEEE Transactions on AES